# Data Access and Dissemination System (DADS) Design Report

July 25, 1996

# **Contents**

Section 1	Overview	1
	Approach	1
	DADS Rapid Development Success Factors	s 2
	DADS Project Planning	3
	DADS Design Stage Deliverables	3
	DADS Project Timeline	4
Section 2	Front-End Modules Design	6
Section 3	Database Design	39
Section 4	Auditing Module Design	48
Section 5	Technical Architecture	56
Appendices		
	Function to Module Matrix	Appendix A
	Database Tables	Appendix B
	Technical Architecture Diagrams	Appendix C

## Section

## **Overview**

1

Census Bureau chartered the development of the Data Access and Dissemination System (DADS) as a part of its increased efforts to facilitate the access and dissemination of official demographic and economic information. This interactive electronic system will be designed to allow efficient and cost -effective access to data generated by the various areas of the Census Bureau.

The DADS project will use an iterative development process, whereby the knowledge gained from each iteration will not be lost, but rather feed into the next iteration. The prototypes will be used to provide customers a way to gain hands-on experience so that they can provide feedback for input into building the actual DADS Production System. These prototypes will also serve to evaluate hardware and software best suitable to access, manipulate, tabulate, and manage very large volumes of data quickly, easily, and cost -effectively. The DADS Production System will be completed in order to serve as the vehicle for accessing and disseminating data derived from the 2000 Census.

The DADS project foresees developing three prototypes leading up to the Production System. The first prototype (Prototype #1) is being developed in-house and is currently underway. An Analysis Report for this prototype was completed and published May 20,1996. This document contains the results of the Design Stage.

## **Approach**

The DADS project has adopted a "rapid development" approach to system development. This type of systems development streamlines and may abbreviate activities in order to reduce the time required from the initial idea to final delivery. However, because of the complexity of the project, a structured approach is still necessary in order to manage an application development effort as large as DADS. The Oracle CASE\*Method is being used as this structured approach. The CASE\*Method consists of a set of stages, tasks, deliverables, and techniques which leads a project through the life -cycle of a system and is especially tailored for developing database applications.

This report is one such deliverable which documents the results of the Design Stage for Prototype #1. It includes the design of the front -end application, the back-end physical database, and the technical architecture.

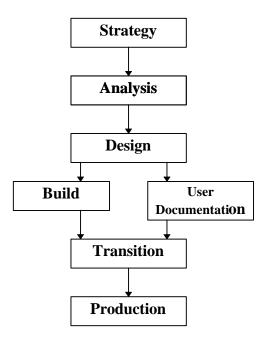


Figure 1. The seven basic stages of CASE\*Method.

## **Rapid Development Success Factors**

Building a team with the right mix of skills is key to the success of the rapid development approach to producing a system. This approach relies on the skills and abilities of the people on the team to work effectively together to maximize time and skill usage. Business awareness, contributed by the DADS management, is also essential to the success of system development.

The skills necessary for success are:

- Business awareness:
  - of general business practices
  - specific knowledge of the organization
- System development skills:
  - problem analysis and problem solving

- specialized computer skills
- generic modeling
- clarity of thought and concept
- Team and project management skills
- Communication skills:
  - within a team
  - between the team and other parties.

Since speed is of the essence in a rapid development project, the team must remain focused on deliverables and goals.

## **DADS Project Planning**

With only six months to develop the first prototype, project kickoff occurred on April 16, 1996, the Analysis Stage was completed May 20, 1996, and the Design Stage July 18, 1996. At the present time, the project is commencing the Build Stage.

In order to manage such a large project with a very tight schedule, DADS invested in structured project planning using Microsoft Project software. This timeline plan follows the CASE\*Method stages, tasks, and deliverables, and includes many concurrent design and development activities. During the Design Stage, the DADS staff members were assigned to two teams -- the front-end and the back-end. Progress was tracked against the timeline plan through regular status reporting and updates to the project plan .

## **DADS Design Stage Deliverables**

During the Design Stage three critical activities occurred:

- 1. The screens (user interface) were designed to reflect and provide required functionality.
- 2. The physical database was designed.
- 3. The technical architecture was further defined based on capacity planning.

The following sections in this report document the three major Design Stage activities and deliverables, namely the Front -end Modules Design, the Database Design (Back-end), and the updated Technical Architecture Report.

## **DADS Project Timeline**

The four major iterations of the CASE\*Method stages will produce three DADS prototypes and finally the production system. The revised overall timeline, after completion of the design stage, is as follows:

## **DADS Prototype #1**

Ready for Beta Testing: Fall 1996

Objective: Provide a "proof of concept" for the basic design,

technology, and functionality of the DADS system.

Provide users the ability to do on-line data

extraction and quick tabulations. Make front-end

application user friendly and test system

performance.

Audience: Internal Users (max 50 concurrent users)

Datasets: 1990 Decennial Census 100% and sample detail

files, PUMS, STF1 & STF3

## **DADS Prototype #2**

Ready for Beta Testing: Fall 1997

Objective: Learn from the first prototype in terms of viable and

scaleable technical architecture, performance tuning, user friendly interface. Expand functionality to all areas of the DADS system. Concentrate on on-line help, ability to create predefined products, and posting these products on the DADS system. Add

functionality to support new datasets.

Audience: Internal and External Users (max 200 concurrent

users)

Datasets: American Community Survey datasets,

1992 Economic Census dataset (test focus is 1997 Econ Census data), all datasets from first prototype.

## **DADS Prototype #3**

Ready for Beta Testing: Fall 1998

Objective: Prepare to fully support the 1998 Decennia Dress

Rehearsal while maintaining support for other datasets and usages. Incorporate feedback from

second prototype.

Audience: Internal and External Users (500+ concurrent users)

Datasets: 1997 Economic Census, 1998 Decennial Census

(dress rehearsal), all datasets from second prototype.

## **DADS Production System**

Ready for Production: Spring 2000

Objective: Provide a fully functional, user-friendly system for

accessing Census data and products from both the

Census intranet and the World Wide Web.

Audience: Internal and External Users (500+ concurrent users)

Datasets: 2000 Decennial Census 100% and sample detail

files<sup>1</sup>, current Economic Census; American Community Survey, and others as needed.

<sup>&</sup>lt;sup>1</sup>The system will be ready in Spring 2000 although the live 2000 data will be loaded at a later date (after processing).

## SECTION 2

# FRONT-END MODULES DESIGN **Bureau of The Census Data Access and Dissemination** System

## **Prepared by DADS Front-End Team**

Author: Albert Lulushi

**Creation Date:** July 12, 1996 Last Updated: July 16, 1996

Version:

1

## Contents

Introduction	8
Overview of the System	9
Log User Subsystem	10
Login	10
Search Criteria Subsystem	12
Search	12
Geography Browser	15
View or Download Products and Reports Subsystem	18
List Products and Reports	18
View Metadata Information	
View Products and Reports	
Download Products and Reports	
View or Download STF Products Subsystem	24
Summary Tape File	24
Extract, Summarize, or Tabulate Data Subsystem	26
List Tables	26
List Data Items	
Specify Calculated Data Items	
Tabulate	
Summarize	32
Count and Compute	
Restrict Record Set	
Download Outcome	36
View Outcome	37

## Introduction

The Front-End Modules Design for Data Access and Dissemination System (DADS) contains a list of all the modules and their associations designed by the front-end development team to implement the functionality of the system. DADS consists of five main subsystems:

- Log User Subsystem
- Search Criteria Subsystem
- Products and Reports Subsystem
- Summary Tape Files Subsystem
- Ad-Hoc Query Subsystem

Each of these subsystems is discussed in this document.

## **Overview of the System**

DADS will allow users to focus their work in five major functional areas. They are presented in the following list.

- Log into the System.
- Specify Search Criteria
- View or Download Products and Reports
- Extract, Summarize, or Tabulate Data from Databases
- View or Download STF Products

Figure 1 shows the relationship between these functional areas.

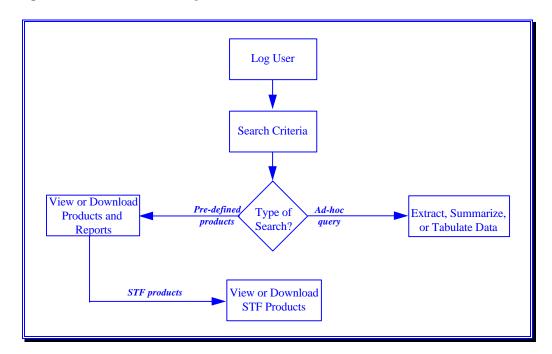


Figure 1: Major Functional Areas of DADS.

Initially users will be required to enter a user name and password in order to access the system. Authenticated users will be able to use all the features of the system. A special GUEST account will allow unregistered users to flavor some of the system's functionality.

The Search Criteria part of the system will enable users to focus on the products and data offered by the Census Bureau that are relevant to them. The search can be restricted in several ways, such as by entering keywords, specifying a subject area, entering the name of a product, or focusing in a geographic area.

Depending on the results of the search, uses will be able to view or download Census' predefined products, or extract, summarize and tabulate data from existing databases.

## **Log User Subsystem**

The purpose of this subsystem is to allow valid users to access the system. The only screen in this system is shown in the following figure.



Figure 2: Logon screen.

## Login

## **Overview**

This screen will allow authorized users as well as visitors to view data and products. Authenticated users will be able to use all the features of the system. New users and guests will be offered a sneak preview of the system.

## **Functionality**

- Identify the user making a connection to the DADS system. (DADS 111)
- Identify the type of user and the data access authority of the user accessing the DADS system. (DADS 112)

## **Inputs and Outputs**

Inputs	Outputs
User Domain (Internal Or External)	Username
Username	Password
Password	Security Level

## **Detailed Description**

• Logon button pressed:

If the Username and Password items are filled in then:

If User Domain is Internal then:

Request connection to internal database; obtain Security Level.

Else

Request connection to external database; obtain Security evel.

End If

Proceed to Search screen.

Else

 $\label{eq:count} Advise \ user \ to \ enter \ Usernam \ e \ and \ Password, \ request \ a \ new \ account, \ or$ 

logon as Guest.

End If

• Guest button pressed:

Request connection to the external database for Guest; obtain Security Level.

Proceed to Search screen.

New User button pressed:

If User Domain is Internal then:

Obtain new user information; create new user account; logon.

Proceed to Search screen.

Else

Advise user to contact the Census Bureau for an account.

End If

## **Search Criteria Subsystem**

The Search Criteria is made up of two principal components: the Search screen and the Geography Browser. The interaction of these components is shown in the following picture.

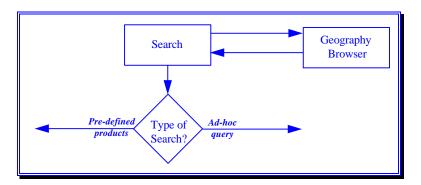


Figure 3: Search Criteria subsystem.

#### Search

#### Overview

The Search screen allows users to specify the type of search they want to perform. It can be for predefined products or an ad-hoc query of the existing datasets. Each search can be restricted to products and databases that contain information about a particular subject area, or that are associated with a keyword. Furthermore, users can search Census products and reports by their type, or by directly specifying their name. Figure 4 shows the Search screen prototype.

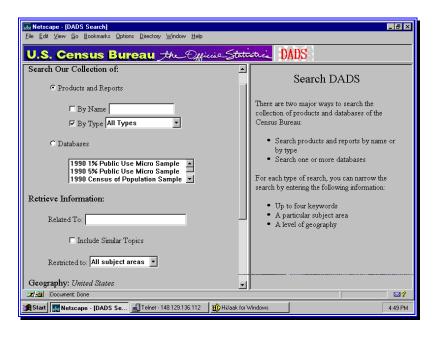


Figure 4: Search screen.

## **Functionality**

- Default: geography search to "US", timeframe search to "most recent info", subject area to "all", product type to "all". Explicitly default data products to those derived from a 100% dataset vs a sample and offer user the choice. (DADS 12211)
- Allow navigation to one or more predefined products by subject area(s). \*Limit to one subject area or all subject areas in 1st prototype. (DADS 12213)
- Allow navigation to one or more predefined products by key word searches. \*Limit up to 4 key words ANDed together in 1st prototype. (DADS 12214)
- Allow navigation to one or more predefined prooducts by product type. \*Limit one product type of all product types in 1st prototype. (DADS 12216)
- Allow user at any time to narrow insterst to geographic areas, subject areas, key words, datasets/surveys and product types to further narrow interests. (DADS 12217)
- Allow "fast track" searches to predefined products for expert users. Search by product name allowing wildcards for pattern matching. (DADS 1222)
- Allow user to tell the system when he/she is ready to submit search for predefined products which meet the user's criteria. (DADS 1223)
- Issue query. (DADS 122331)
- Provide option to cancel current search and querying activity at anytime. (DADS 12234)
- Check for dataset availability and only allow access to datasets which are currently available and for which the user has authorized access (e.g. to Title 13 datasets). Display unavailable/unauthorized datasets in dim and explain. (DADS 13211)
- Default geographic area to US. Default timeframe to current. Default dataset for external and internal users unless subject area or keyword search is used or another dataset is selected. (DADS 13212)
- Allow user to limit datasets and records queried via limiting geographic areas. (DADS 13213)
- Allow user to limit datasets queried via limiting subject areas. (DADS 13215)
- Allow user to limit datasets queried via keyword searches. Note tables with data field and valid value matches to keywords for later selection list query bui.lding. (DADS 13216)
- Allow user to select a dataset from a list of datasets available to the user. (DADS 13217)
- Allow user to perform searches from: geography, timeframe, subject area, and keywords OR from: geography and specific dataset(s).. (DADS 13218)

#### **Inputs and Outputs**

Inputs	Outputs
Username	Username
Password	Password
Security Level	Security Level

Prior Screen Prior Screen

Product Category Indicator Product Category Indicator

Predefined Product Name Predefined Product Name

Predefined Product Type Predefined Product Type

Dataset Name Dataset Name

Keywords String Keywords String

Synonym Search Indicator Synonym Search Indicator

Subject Area Subject Area

Geography Codes String Geography Codes String

Current Geography Level Current Geography Level

Current Geography Value Current Geography Value

## **Detailed Description**

Build Search screen:

Retrieve predefined products types, available datasets, and subject areas from database.

If Prior Screen is Geography Browser then

Adjust screen controls according to the values of the input variables Product Category Indicator through Current Geography Value.

Else

Set Current Geography Level to US and Current Geography Code to US

End If

Geography Search hyperlink clicked:

Proceed to Geography Browser screen.

Search button pressed:

If Product Category Indocator is Predefined Products then

Proceed to List Products and Reports screen.

Else

Proceed to List Tables screen.

End If

## **Geography Browser**

#### **Overview**

Another way to restrict the search of products, reports, and databases is by narrowing down the geography scope of the search. Users will use the Geography Browser shown in Figure 5.

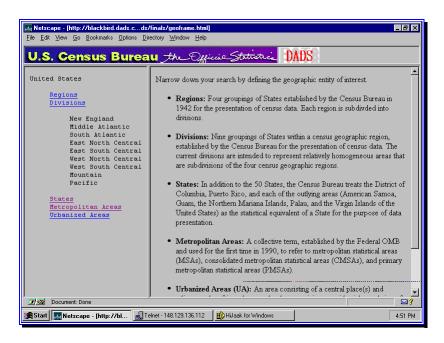


Figure 5: Geography Browser screen.

The geography browser will present the geography in a hierarchical fashion, starting from the highest level, the United States, down to the lowest levels, blocks and block groups. Users will be able to view and select a component within a geography level, for example Fairfax County at the County level; or they will be able to expand a given geography level to its sub-levels, for example County to MCD's/CCD's, and Census Tracts/BNA's.

## **Functionality**

- Allow navigation to one or more predefined products by geography where applicable. (DADS 12212).
- Allow user at any time to narrow insterst to geographic areas, subject areas, key words, datasets/surveys and product types to further narrow interests. (DADS 12217)
- Allow user to limit datasets and records queried via limiting geographic areas. (DADS 13213)
- Allow user to perform searches from: geography, timeframe, subject area, and keywords OR from: geography and specific dataset(s).. (DADS 13218)

## **Inputs and Outputs**

Inputs Outputs

Username Username

Password Password

Security Level Security Level

Prior Screen Prior Screen

Product Category Indicator Product Category Indicator

Predefined Product Name Predefined Product Name

Predefined Product Type Predefined Product Type

Dataset Name Dataset Name

Keywords String Keywords String

Synonym Search Indicator Synonym Search Indicator

Subject Area Subject Area

Geography Codes String Geography Codes String

Current Geography Level Current Geography Level

Current Geography Value Current Geography Value

#### **Detailed Description**

• Build Geography Browser screen:

Based on Geography Codes string, expand hierarchy tree to Current Geography Level and Current Geography Value.

Retrieve list of Geography Objects where Parent Object is Current Geography Level.

Folder Icon clicked:

If Object is Geography Level name then

Set Current Geography Level to the clicked geography level

Append Current Geography Level to Geography Codes string.

Else

Append Current Geography Value to Geography Codes string.

## End If

Proceed to Geography Browser screen.

• Geography Instance name hyperlink clicked:

Set Current Geography Value to the clicked geography instance

• Search button pressed:

Append Current Geography Value to Geography Codes string.

Return to Search Screen.

## **View or Download Products and Reports Subsystem**

This subsystem is made up of the following components:

- List Products and Reports screen
- View Metadata Information screen
- View Products and Reports screen
- Download Products and Reports screen

The interaction between these components is shown in Figure 6.

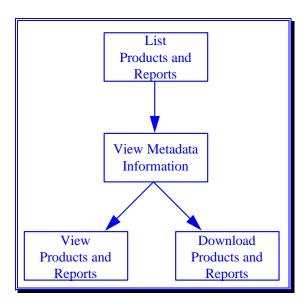


Figure 6: Products and Reports subsystem.

## **List Products and Reports**

## Overview

This screen will display a list of products and reports that match the search criteria specified by the users in the Search subsystem. A sample of this screen is shown in Figure 7.



Figure 7: List Product and Reports screen.

This screen will display 15 products at a time, and will show the name, type, size, and format of the product. By clicking the product name, users will be able to navigate to View Metadata Information screen.

## **Functionality**

- Issue query and return the count of how many predefined products meet the user's criteria. (DADS 12231)
- Provide option to further define query and re-execute within the general or expert track. (DADS 12232)
- Retrieve list of predefined products. Display 15 records at a time and allow user to scroll through the list and view/download selected predefined products. (DADS 122332)

#### **Inputs and Outputs**

Inputs Outputs
username selected product
password
search parameters

#### **Detailed Description**

After you select a product it makes database queries to collect the metadata for that product and creates the metadata page.

## **View Metadata Information**

#### **Overview**

This screen will display metadata information about the product or report, such as title, description, instructions on how to use it, and so on. After reviewing the metadata information, users may select to either view or download the product or report by clicking the appropriate button. Figure 8 shows the View Metadata Information screen.

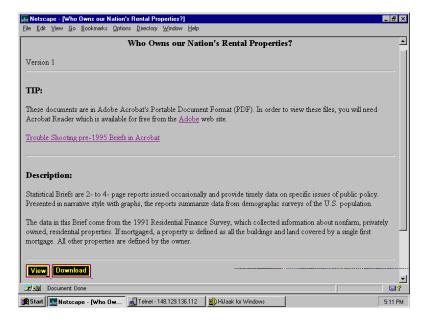


Figure 8: View Metadata Information screen.

## **Functionality**

- Based on the product format (e.g. HTML, GIF, ASCII, ACROBAT, etc.), determine if viewable and how best to view. (DADS 1231)
- Display predefined product. (DADS 1232)
- Check for dataset availability and only allow access to datasets which are currently available and for which the user has authorized access (e.g. to Title 13 datasets).
   Display unavailable/unauthorized datasets in dim and explain. (DADS 13211)
- Allow a user to select a dataset from a list of datasets available to the user. (DADS 13217)

#### **Inputs and Outputs**

Inputs	Outputs
username	password
password	username
product id	product id

## **Detailed Description**

 Get input parameters and fetch the name and description for the selected product from the following cursor.

SELECT D.SHORT\_NAME, D.LONG\_NAME, D.LONG\_DESCR, P.DESCRIPTION FROM DADS\_PRODUCTS D, PREDEFINED\_TYPES P
WHERE D.ID = :PRODUCT\_ID
AND P.PREDEFINED\_TYPE = D.PREDEFINED\_TYPE

• Build HTML page with results (singleton-select).

```
<TITLE>short_name</TITLE>
<H3>long_name</H3>
<H5>TIP:</H5>description
<H5>DESCRIPTION:</H5>long_description
```

## **View Products and Reports**

#### **Overview**

This screen will display the selected product or report in a format that is appropriate for viewing. Figure 9 shows an example of a product saved in the form of an image file.

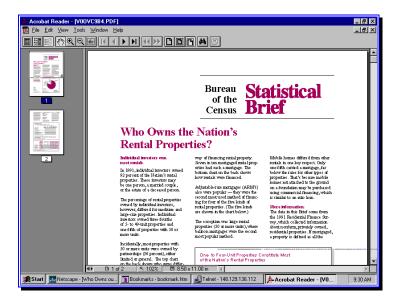


Figure 9: View Products and Reports screen.

## **Functionality**

• Display predefined product. (DADS 1232)

#### **Inputs and Outputs**

Inputs Outputs
username username
password password
product url

## **Detailed Description**

The user will see the data displayed at the existing URL somewhere on the Census site. This is the end of the query session on the DADS system, unless we include a link back to the start of a new query. In that case, the user's name will be an output from this page.

## **Download Products and Reports**

#### **Overview**

This screen will allow users to download the chosen product or report. It will provide information and tools that will facilitate the process of downloading. For example, users will be able to download compressed files to reduce download time, and decompressing software will be provided for them. Users will also be notified about estimated download time based on the product size and network or modem speed. Figure 10 shows a prototype of the Download Products and Reports screen.

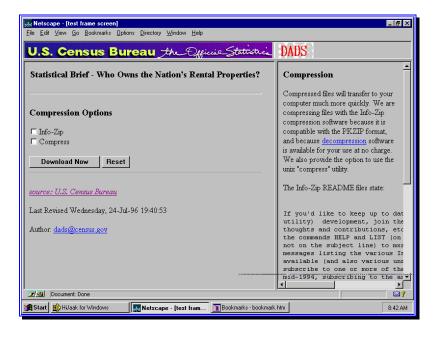


Figure 10: Download Products and Reports screen.

#### **Functionality**

- Based on the product format (e.g. HTML, GIF, ASCII, ACROBAT, etc.), determine if viewable and how best to view. (DADS 1231)
- Display predefined product. (DADS 1232)
- Provide download destination options. (DADS 1241)

- Provide option of compressing files before downloading, and provide realted decompression software for downloading. (DADS 1242)
- Estimate time to download and notify the user. (DADS 1243)
- Allow user to cancel download request at anytime before or during the download. (DADS 1244)
- Notify user of status of download. (DADS 1245)

## **Inputs and Outputs**

Inputs Outputs

username username

password password

predefined products compression choice

location on DADS

## **Detailed Description**

The user is presented with several choices for the optional compression format for their output file. The choices that the user makes will be outputs from this page for the backend program/script to prepare the data product.

## View or Download STF Products Subsystem

Users will be able to view and download information from the Summary Tape Files using the View and Download STF Products screen shown in Figure 11.

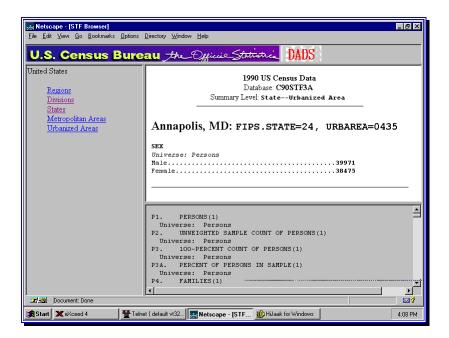


Figure 11: View and Download STF Products screen.

## **Summary Tape File**

## Overview

The STF browser is powerful because it allows users to view STF products in one screen by applying bi-directional searches. By picking a geography, users can view any STF table for that geography by selecting the table name in the bottom-right pane.. By picking a STF table, users can view this table for any geography by specifying the geography in the left pane.

#### **Functionality**

- Issue query and return the count of how many predefined products meet the user's criteria. (DADS 12231)
- Provide option to further define query and re-execute within the general or expert track. (DADS 12232)
- Retrieve list of predefined products. Display 15 records at a time and allow user to scroll through the list and view/download selected predefined products. (DADS 122332)

#### **Inputs and Outputs**

Inputs Outputs

username	username
password	password
table	
summarylev	
geocode	

## **Detailed Description**

Upon entering this screen, the user will be presented with three views: a geography navigator, a list of table titles and the current table selection. Using the geography navigator, a user can change geography for the current table. Likewise a user can change the current table by making a selection off the table list.

- geobrowse.pl: The PERL script to control the look and function of the geography navigator.
- stfbrowse.pl: The PERL script to process request and generate table view.

Example API: ./cgi-bin/stfbrowse.pl?TABLE=P0;SUMLEV=050;GEOCODE=24.027

Logic:
Parse command line.
Lookup SUMLEV.
Parse GEOCODE.
Issue SQL Query.

Send formatted results to table frame.

## **Extract, Summarize, or Tabulate Data Subsystem**

This subsystem is made up of the following components:

- List Tables screen
- · List Data Items screen
- Specify Calculated Data Items screen
- Tabulate screen
- Summarize screen
- Count and Compute screen
- Restrict Record Set screen
- View Outcome screen
- Download Outcome screen

The interaction between these components is shown in Figure 12.

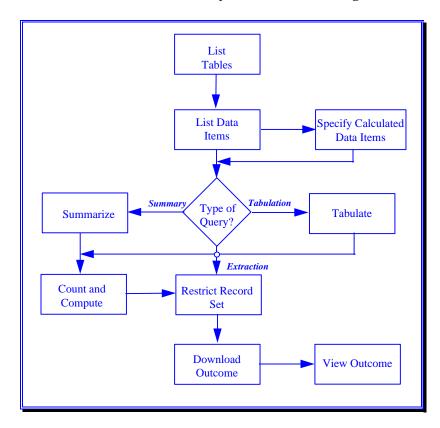


Figure 12: Extract, Summarize, and Tabulate subsystem.

## **List Tables**

#### **Overview**

This screen will display the databases and corresponding tables that match the search criteria the users specified in the Search subsystem. Users may pick one or more tables from a database in order to view the data they contain. The data items in these tables are

displayed in the List Data Items screen, which follows this screen. Figure 13 shows a prototype of List Tables screen.

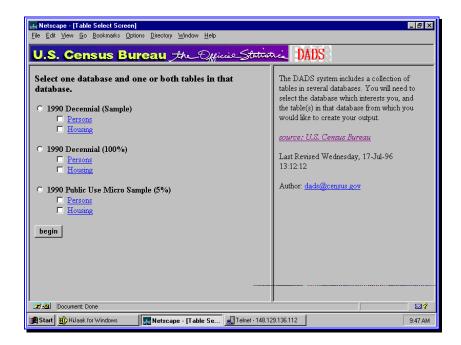


Figure 13: List Tables screen.

## **Functionality**

- Significance/meaning of ad-hoc query results. (DADS 1324)
- Provide metadata for building queries for user observation and selection. (DADS 1331)
- Highlight results of navigational searches: datasets selected, and tables and columns selected through keyword searches (if any), and geographic conditions. (DADS 13311)
- Show the hierarchical mapping of selected datasets to objects to columns. (DADS 13312)
- Help user to select table(s) of interest to query from selected dataset(s) and create
  Provide disclaimer of statistical validity of any user's ad-hoc query or the the
  "FROM clause." Warn user when joining tables across datasets. (DADS 133221)

## **Inputs and Outputs**

Inputs	Outputs
username	database selected
password	tables selected

#### **Detailed Description**

For the dataset and tables selected display all fields in the field selection page.

## **List Data Items**

#### Overview

This screen allows users to select one or more data items to include in the output of the query, or to use for restricting the outcome of the query. If users want to include calculated data items in the query, they can define them to the screen Specify Calculated Data Items.

After selecting the data items, users choose the type of query to execute with these data items. Clicking Summary will take them to the screen Summarize; clicking Table will take them to Tabulate; and clicking Extract will take them to the screen Restrict Record Set. The List Data Items screen is shown in Figure 14.

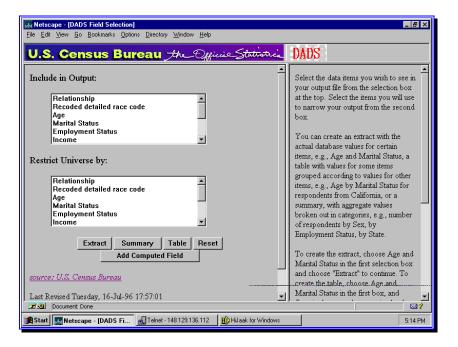


Figure 14: List Data Items screen.

#### **Functionality**

- Provide metadata for building queries for user observation and selection. (DADS 1331)
- Show the hierarchical mapping of selected datasets to objects to columns. (DADS 13312)
- Make user select type of query: customized extraction at record level, or statistical summarization which would include tabulations and the return of one summarized record or value. (DADS 1332)
- Help user build the "SELECT clause." Allow selection of any column and allow support for ratios between columns and other calculations. (DADS 133213)
- Help user to build the "SELECT clause." Support use of any group function. (DADS 13224)

#### Inputs and Outputs

Inputs Outputs
username username
password password
table\_list select\_list
where\_list

## **Detailed Description**

 Get input parameters and fetch the table name, column name and description for the selected items from the following cursor.

```
SELECT TABLE_NAME, COLUMN_NAME, SHORT_DESC FROM DB_TABLE_COLUMNS

WHERE TABLE_NAME = :TABLE_LIST-item

[ OR TABLE_NAME = :TABLE_LIST-item ]...
```

• Fill HTML list boxes with these values.

```
<SELECT NAME="SELECT_LIST" MULTIPLE SIZE=6>
     <OPTION VALUE="table_name.column_name">short_desc
</SELECT>

<SELECT NAME="WHERE_LIST" MULTIPLE SIZE=6>
     <OPTION VALUE="table_name.column_name">short_desc
</SELECT>
```

## **Specify Calculated Data Items**

#### **Overview**

This screen will enable users to include calculations of numeric data items in their queries, such as the ratio between the income of householder and the total household income. Suggested operations will be the four basic arithmetic operations.

After selecting the data items, users choose the type of query to execute with these data items. Clicking Summary will take them to the screen Summarize; clicking Table will take them to Tabulate; and clicking Extract will take them to the screen Restrict Record Set. The Specify Calculated Data Items screen is shown in Figure 15.

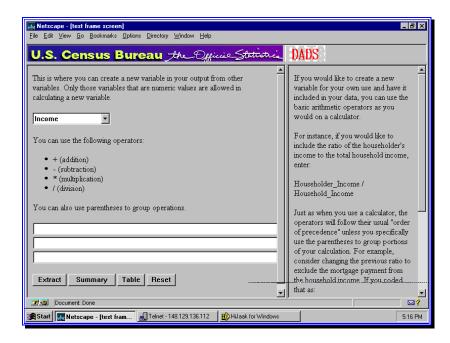


Figure 15: Specify Calculated Data Items screen.

## **Functionality**

- Provide metadata for building queries for user observation and selection. (DADS 1331)
- Make user select type of query: customized extraction at record level, or statistical summarization which would include tabulations and the return of one summarized record or value. (DADS 1332)
- Help user build the "SELECT clause." Support use of any group function. (DADS 133224)

## **Inputs and Outputs**

Inputs	Outputs
username	username
password	password
colnames choose by user	output cols based on stat. formulae

## **Detailed Description**

The user is presented with an opportunity to apply several arithmetic operators on the data from the columns which they have included in their data request. Only columns whose data is numeric (as determined by the DADS meta-data tables) will be subject to arithmetic manipulation.

The user can construct a calculated field using addition, subtraction, multiplication, division, or explicit grouping (to implement a non default precedence of operation).. The user is responsible for producing a usable data product. DADS' online documentation

will provide a sample of how to best code particular types of calculated data columns. Four pushbuttons ("SUBMIT") at the bottom of the page allow the user to choose between producing an extract of data, a summary of data (i.e., a single value for some subset of the Census data), a table (i.e., a collection of values of some data item(s), ordered by the value(s) of some other data item(s)), or to clear all inputs on this page and begin working on this page again.

#### **Tabulate**

#### **Overview**

This screen will allow users to summarize their data in the form of two-dimensional repeated for each value of a third variable. For example, users may want to compute the number of students for each School Enrollment category, for each Citizenship type, and for each county in their state. The screen will show users the picture of a sample table, and will allow them to pick the data items for the dimensions of the tables. After this screen, users will navigate to the screen Count and Compute in order to specify the kind of computation they want to place in the cells of the table. This screen is shown in Figure 16.

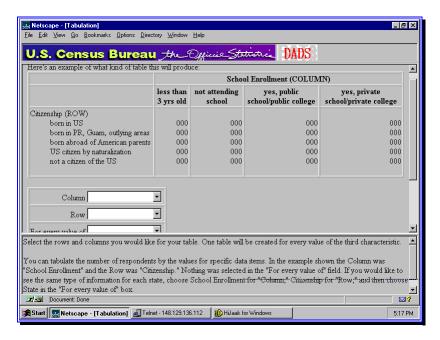


Figure 16: Tabulate screen.

#### **Functionality**

- Provide metadata for building queries for user observation and selection. (DADS 1331)
- Help user determine group by column(s), and create the "GROUP BY clause". (DADS 133223)

## **Inputs and Outputs**

Inputs Outputs

username username
password password
dataset selected database selected
tables selected tables selected
fields selected column
row
page

## **Detailed Description**

Retain dataset, tables, column, row and page and create values page.

#### **Summarize**

#### **Overview**

This screen will allow users to summarize their data by up to five variables. This will enable them, for example, to compute the average income of citizens by sex, by race, by marital status, by county for a give state. The screen will show users the picture of a sample record set, and will allow them to pick the data items for the dimensions of the summary. After this screen, users will navigate to the screen Count and Compute in order to specify the kind of computation they want to place in the value cells of the summary table. This screen is shown in Figure 17.

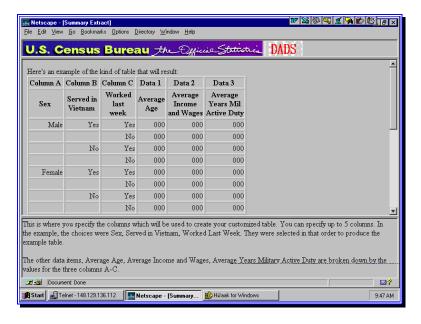


Figure 17: Summarize screen.

## **Functionality**

 Provide metadata for building queries for user observation and selection. (DADS 1331)  Help user determine group by column(s), and create the "GROUP BY clause". (DADS 133223)

## **Inputs and Outputs**

Inputs	Outputs
username	username
password	password
database selected	database selected
tables selected	tables selected
fields selected	dimensions 1-5

## **Detailed Description**

Retain dataset, tables, dimensions and create functions page.

## **Count and Compute**

#### Overview

Users will navigate to this screen only if they are performing tabulations or summaries. Here they will specify the function to apply to those data items they want included in the outcome of the query. Users will be able to count all the data items, and for the majority of the data items, this will be the only allowed operation. For numeric data items, such as age, or householder income, users have the ability to compute other operations, such as average, minimum, and maximum. The Count and Compute Screen in shown in Figure 18.

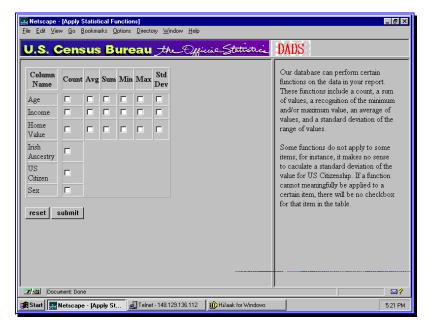


Figure 18: Count and Compute screen.

## **Functionality**

- Provide metadata for building queries for user observation and selection. (DADS 1331)
- Help user determine group by column(s), and create the "GROUP BY clause." (DADS 133224)

## **Inputs and Outputs**

Inputs

Outputs

username

username

password

varname

colname

function

## **Detailed Description**

Upon the selection of the statistical functions option from the ad-hoc query page, the user is placed on this page, the statistical functions selection page, where the user selects from the available descriptive statistics functions i.e count, sum, avg, min, max values presented in the check boxes.

## **Restrict Record Set**

## **Overview**

This screen will allow users to restrict the outcome of their queries by entering criteria on the data items they picked in screen List Data Items. The Restrict Record Set screen is shown in Figure 19. It displays all the possible values for the selected data items, thus enabling users to specify their restriction criteria by clicking on the desired values. In the scenario presented in Figure 19, for example, users may restrict their record set to persons who are householders or their spouses and between 20 and 45 years of age by selecting the first two items in the list Relationship\_to\_householder, by choosing *between* from the list under Age, and by entering values *20* and *45* in the text items to its right.

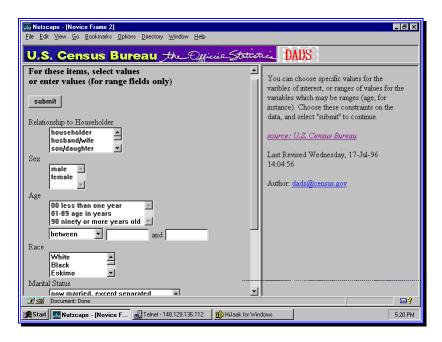


Figure 19: Restrict Record Set screen.

#### **Functionality**

- Provide metadata for building queries for user observation and selection. (DADS 1331)
- Allow user to set any conditions in addition to geography conditions to limit records returned with AND and OR operands between clauses, and create the "WHERE clause." (DADS 133212)
- Allow user to set any conditions in addition to geography conditions to limit records returned with AND and OR operands between clauses, and create the "WHERE clause." (DADS 133222)

#### **Inputs and Outputs**

Inputs	Outputs
username	username
password	password
extract	values
table	

#### **Detailed Description**

Upon the selection of an extract or table option from the ad-hoc query page, the user is placed on this page, the selection criteria page, where the user selects from the available values presented in the list boxes to build a user-defined query.

#### **Download Outcome**

#### Overview

This screen will be displayed after users have specified the criteria to restrict their record set. It offers them the opportunity to title the query, to restrict the number of records returned, and to display the query on the screen or write the outcome to a file. If users choose to view the query, the View Outcome screen will appear. If they want to write it to a file, they can select a format for the file and choose other options such as to include the data items' headers, to compress the file, and so on. The Download Outcome screen is shown in Figure 20.

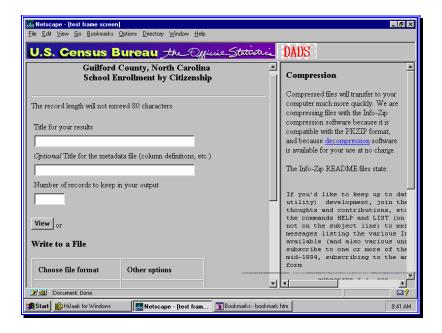


Figure 20: Download Outcome screen.

#### **Functionality**

- Based on the product format (e.g. HTML, GIF, ASCII, ACROBAT, etc.), determine if viewable and how best to view. (DADS 1231)
- Display predefined product. (DADS 1232)
- Provide download destination options. (DADS 1241)
- Provide option of compressing files before downloading, and provide realted decompression software for downloading. (DADS 1242)
- Estimate time to download and notify the user. (DADS 1243)
- Allow users the option of producing a metadata file associated with the user's query for viewing/downloading. (DADS 13313)
- Allow user to refine query after execution and use to change types of query. (DADS 1333)
- Allow user to tile the query output. (DADS 1351)
- Support storing just the data results without column headings for loading data into other packages. (DADS 13531)
- Notify user of maximum record length of query output. (DADS 1354)

- Allow user to select the number of records to be returned in total. (DADS 1356)
- Provide downloading capabilities. (DADS 137)
- Allow user to select output formats. (DADS 1353)

#### **Inputs and Outputs**

Inputs Outputs

username username

password password

extract name in DADS title for data file

user's printer driver title for metadata file

compression choice

file format

user's printer name (optionally greyed out)

#### **Detailed Description**

The user is presented with several choices for the file format and the optional compression format for their output file. The user will also choose any title(s) for the data file and/or its accompanying metadata file. The user will be allowed to title the metadata file separately from the data file, and will be able to include the SQL statements which generated the data outputs in the metadata file. All of these options apply to files which the user will download to their computer. The choices that the user makes will be outputs from this page for the backend program/script to prepare the data product.

The user is given the choice to view the output of the query. This option is represented by a "submit" button which will send the request to the backend script to display the SQL results in a text form.

#### **View Outcome**

#### **Overview**

This screen will be displayed if users choose to display the outcome of their query to the screen. Figure 21 is a prototype of the View Outcome screen.

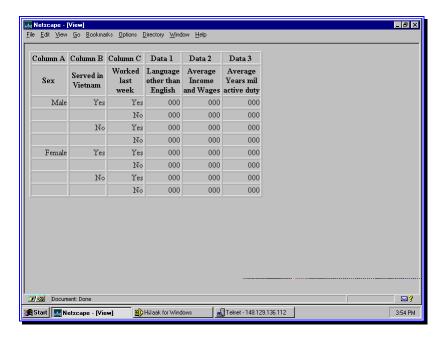


Figure 21: View Outcome screen.

#### **Functionality**

- Allow user to refine query after execution and use to change types of query. (DADS 1333)
- Offert types of display formats for on-line query results. Allow users to choose not to display query results. (DADS 134)
- Estimate time to perform the query and notify the user. Allow user to execute query or cancel. Potentially allow the system to refuse execution if the cost it too high. (DADS 1362)
- Allow user to cancel the query at any time. (DADS 1364)

#### **Inputs and Outputs**

Inputs	Outputs
username	username
password	password
<b>Detailed Description</b>	

#### SECTION 3

# DATABASE DESIGN Bureau of The Census Data Access and Dissemination System

#### **Prepared by DADS Back-End Team**

Author: Gail Wright
Creation Date: July 17, 1996
Last Updated: July 17, 1996

Version: 1

#### **Physical Database Design**

The first DADS prototype will access three Census datasets and two supportive datasets. These datasets are:

- The 1990 Decennial Dataset
- The 1990 5% Public Use Microdata Sample (PUMS) Dataset (possibly 1% PUMS also)
- The 1990 Summary Tape Files (STF1 and STF3)
- The underlying geography associated with the datasets
- Supportive data about the datasets and predefined products

These datasets were logically modelled during the Analysis stage of the DADS project. The Oracle Designer/2000 software was used to model these datasets in what are called Entity Relationship Diagrams. (See the *DADS Analysis Report* for more information on data modelling.) During the Design stage, the information from the Analysis models is converted into a physical database architecture. This conversion was performed by running a Designer/2000 utility. In general, entities become database tables and relationships become database fields used to join tables together. All dataset data will be stored in the Oracle7 database in the first prototype.

NOTE: All physical database tables are documented in Appendix B.

#### **Designing the Physical Database for Performance**

It is during the Design stage that performance considerations are made and the physical database structure is architected for its intended use. For most OnLine Transaction Processing (OLTP) systems and small databases, the default physical database structure which parallels its logical model works very well. However, for large data warehousing applications, special tuning considerations can impact the physical database structure. This applies especially to the 1990 Decennial Dataset which is a read-only database storing approximately 150 gigabytes of data.

The technology for architecting data warehouses is changing everyday. The ultimate performance challenge is being able to process ad-hoc queries against a very large database quickly. The problem ad-hoc queries add to the performance challenge is the problem of the unknown. If the database designer knew what queries an application written against the data would issue, the designer could tune the database for its specific use. However, if its use is somewhat unknown, then the designer is not able to favor one type of query versus another.

The primary performance problem of large data warehouses is I/O throughput and contention. Computer technology has made great strides in increasing CPU speeds and lowering the cost per MIPS (millions of instructions per minute), unfortunately however, advances in I/O throughput

haven't been as dramatic. Thus the software world is trying to address the I/O-bounded problem which is especially made apparent by data warehousing applications. The basic solution is to reduce the physical I/O needed to provide a user with the appropriate response to his/her request. This can be accomplished in a number of ways including:

#### Partitioning Database Tables

Tables can be partitioned *horizontally* (where records of a logical entity are separated into many tables based on some business "dimension") and/or *vertically* (where columns of a logical entity are separated into different tables along with the primary key needed to join the vertically partitioned tables back together). Thus a logical entity is "chunked up" into several smaller tables where the hope is that to answer a user's request only one partitioned table or a few such tables need to be searched. If in general many of these tables need to be searched, then partitioning does not offer performance advantages.

For the first prototype, the Decennial entities (sample persons, sample housing, 100% detail persons, and 100% detail housing) will be partitioned both horizontally and vertically. These entities will be partitioned horizontally via the state geography dimension (e.g. one sample persons table per state). The allocation flags for each entity will also be stored in separate tables, causing a vertical partition. The hope is that users will generally query data at a state or lower geographic level and will rarely select allocation flags for viewing or searching.

#### Pre-aggregating Data

Whenever a user requests a "cross tabulation", a count or sum or average or another group function, data is being searched, sorted, grouped, and summarized (or in other words, *aggregated*), unless, of course, the user queries *pre-aggregated* data. The Summary Tape File (STF) dataset is an example of pre-aggregated data. Because the data is summarized, there is less of it to search causing less I/O. Additionally, less CPU time is needed as this work took place when the data was first aggregated.

However, there are many issues involved in preaggregating data. Data is aggregated over one or more dimension-levels such as time-year, month, week; geography-US, region, state, county; age; race; sex; etc. where time and geography have multiple levels, but age, race, and sex do not. For the first prototype, there are 18 geographic levels which are associated with persons and housing. 100% Detail Persons has 6 additional dimensions. 100% Detail Housing has potentially 23 additional dimensions. Sample Persons and Sample Housing have more dimensions than you'd want to count. To create pre-aggregated data for every possible tabulation is unthinkable. It would require more space than to store the unaggregated data and a lot of programming to figure out what tables to query. Current thought is that one should not preaggregate every possible combination, but should be "practical" and aggregate only what makes

sense from a business application perspective. However, when the application is an Internet/Intranet ad-hoc query of Census data, knowing what aggregations a person will request is difficult.

Therefore, initially the first prototype will support the pre-aggregated STF dataset for canned (predefined reports) and ad-hoc query, as well as the unaggregated 1990 Decennial dataset for both data extraction and tabulation ad-hoc query. One fear is that users will query from the Decennial dataset for tabulations which could be produced much faster by querying the STF dataset. Future prototypes should be able to address this situation. Also, by auditing the first prototype usage, we may better know what kinds of pre-aggregation is needed.

#### Accessing Small Indexes

Very small indexes can be created against a column(s) in a large table which are then used to process a user's request instead of accessing the large table itself. Oracle and other vendors are using a new indexing technology called "bitmapped indexing" to do just that. Bitmapped indexes are especially useful when searching or grouping by low cardinality fields such as sex, marital status, race, etc. In the past, RDBMs would perform full table scans which are much slower due to the high amount of I/O needed to process the request.

For the Decennial dataset, bitmapped indexes are created on each low cardinality field. Concatenated bitmapped indexes are also created for use when multiple fields are used in a group by clause. Current technology restricts concatenated bitmaps to 16 fields. Since more than 16 low cardinality fields exist in many tables and it is not known which fields will be grouped together, best guesses were made about popular groupings. Future Oracle technology should eliminate the need for concatenated indexes which should have a positive performance impact in future prototypes.

#### • OLAP, MultiDimension, and Other Technologies

A number of technologies are emerging to address the large data warehousing access problem. OLAP (OnLine Analytical Processing) technology supports DSS (Decision Support System) applications which access large read-only data warehouses for the purposes of analyzing the data and potentially making management decisions from such analysis. With OLAP technology, data is stored in a multidimension structure in the RDBMS. New OLAP methodologies and software help the database designer to pre-aggregate data in an RDBMS and help application developers to access this data. OLAP software can be client application software, middleware, and/or software which is tightly integrated with the RDBMS. The concept of partitioning, aggregating, and indexing data via business dimensions (multidimension) is central to the OLAP technology.

Every month new data warehousing technologies are emerging and being discussed in professional journals. Currently, the business application need for these technologies has slightly outpaced a stable and optimal technical solution, thus the DADS project is on the "bleeding edge" of technology. The DADS team is following current technological advances and will likely implement new data warehousing solutions as they become viable.

#### The 1990 Decennial Dataset

#### The Data Source

The CENSAS dataset will be used to populate the Decennial tables in the first prototype. The June 1996 version of the *CENSAS User Handbook* documents new recode fields for Sample Persons and Sample Housing. The DADS team intends to use this updated dataset and include the new recode fields.

#### Physical Database Architecture

The 1990 Decennial Dataset consists of four logical entities:

- 100% Detail Persons
- 100% Detail Housing
- Sample Persons
- Sample Housing

Each entity is partitioned horizontally by state geography (or equivalent for Puerto Rico, etc. - if data exists for these equivalents). Supposing there are only 50 states (without equivalents), this means that the four entities would be mapped to 4\*50=200 physical database tables. Furthermore, each table is vertically partitioned to separate allocation flags from the rest of the fields. Hence the number of physical database tables for storing the Decennial dataset is:

4(entities) \* 2 (vertical partitions) \* x(states and equivalents)

Table names will be in the format:

PERSONS100\_<FIPS State Code>

A\_PERSONS100\_<FIPS State Code>
HOUSING100\_<FIPS State Code>

A\_HOUSING100\_<FIPS State Code>

A\_HOUSING100\_<FIPS State Code>

PERSONS\_SAMPLE\_<FIPS State Code>

A\_PERSONS\_SAMPLE\_<FIPS State Code>

A\_PERSONS\_SAMPLE\_<FIPS State Code>

-- 100% Detail Persons Allocation Flags
-- 100% Detail Housing Allocation Flags
-- Sample Persons
-- Sample Persons Allocation Flags

HOUSING\_SAMPLE\_<FIPS State Code> -- Sample Housing
A\_HOUSING\_SAMPLE\_<FIPS State Code> -- Sample Housing Allocation Flags

Column names will be restricted to 8 characters and will follow the naming convention established in by PUMS where applicable. Although Oracle allows 30 character columns, the design team chose to use 8 characters for a number of reasons including: easier typing and less chance of errors for developer code and end-user queries and easy conversion to SAS datasets. A 40 character description of each column will be presented to the user along with the 8 character column name for easy recognition.

100% Detail Persons can be joined to 100% Detail Housing, and Sample Persons can be joined to Sample Housing. Allocation Flag tables can be joined back to their respective master tables. However, sample tables cannot be joined to 100% detail tables. For this reason, the 1990 Decennial "dataset" will be treated as two separated "databases" when presented to the user: the 1990 Decennial 100% Detail Database and the 1990 Decennial Sample Database.

The first prototype will associate the following geography with both persons and housing:

- State
- County
- Census Tract
- Block Group
- Block
- County Subdivision
- Place
- Metropolitan Statistical Area
- Consolidated Metropolitan Statistical Area
- Urbanized Area
- 101st Congressional District
- 103rd Congressional District
- American Indian Reservation
- Alaska Native Regional Corporation
- Zip Code
- Rural or Urban Area Indicator

The codes of each geographic area listed above will be stored in the persons and housing tables (except allocation flag tables) for quick access of geographic level searches.

Database views will be created for the US, Region, and Division level searches. These views will union the applicable State level tables. NOTE: It is possible that for performance reasons, that only the PUMS dataset will be made available for aggregation/tabulation above the state level.

#### **Security**

In the first prototype, the 1990 Decennial dataset will only be made available internally to the Census by placing this dataset on the Census Intranet inside of its firewall. It is assumed that any user who can access this system is a sworn Census employee or contractor. The system will be username and password protected only for the purpose of controlling the number of beta test users. Future prototypes may include some type of disclosure avoidance which allows making the Decennial dataset available to unsworn users on the Internet.

#### The 1990 PUMS Dataset

The 5% Public Use Microdata Sample (PUMS) data will be made available for ad-hoc query both to internal Census beta testers and to external (Internet) DADS beta testers. The PUMS data is similar to the Decennial data except that there is less of it and it is edited for public use. The physical database architecture should follow the basic Decennial architecture. Benchmarking which is planned for the coming month should prove the best architecture for both Decennial and PUMS datasets. The DADS application will access this dataset the same way it accesses the Decennial dataset. The DADS functionality will be identical regardless of which dataset is selected.

#### The 1990 Summary Tape File Datasets (STF1 and STF3)

The STF dataset will also be made available to both internal and external beta testers. The STF dataset is pre-aggregated 1990 Decennial data. It can be accessed as canned reports (similarly to how it is currently accessed in the Census Lookup application on the Internet), or it can be queried in an ad-hoc environment. The STF dataset was logically modeled during the Analysis Stage, but as of yet the physical database architecture has not been designed.

#### The 1990 Geography Dataset

#### The Data Source

1990 Geographic data is used to maintain geographic consistency between datasets (all timestamped 1990 in the first prototype). The source for the geography data comes from:

1) the Geographic Boundary files, 2) the Geographic Identity Code Scheme CD, and 3) CENSAS geography data.

#### Physical Database Architecture

For each of the 18 geographic entities which we care about for the first prototype (except for the US level), a physical table is created to store the identifying codes, names, types, sizes, etc. of geographic objects (as applicable and available). The geographic tables are as follows:

- REGIONS
- DIVISIONS
- STATES
- COUNTIES
- CENSUS TRACTS
- BLOCK\_GROUPS
- BLOCKS
- COUNTY\_SUBDIVISIONS
- PLACES
- METROPOLITAN\_STATISTICAL\_AREAS
- CONSOLIDATED MSAS
- URBANIZED\_AREAS
- CONGRESSIONAL DISTRICTS 101
- CONGRESSIONAL DISTRICTS 103
- AMERICAN INDIAN RESERVATIONS
- ALASKA NATIVE REGIONAL CORPS
- ZIP\_CODES

The FIPS codes are used as the primary keys where applicable (instead of using Census codes). The column names of identifying geographic entities are used consistently throughout all datasets when used as primary or foreign keys and are restricted to 8 characters. Foreign keys are created but not enabled. This allows database level documentation in the Oracle data dictionary, and possible use by the application without any performance degradation on load and update.

Two supportive geographic tables are also maintained as follows:

- GEO\_OBJECTS -- maintains list of geographic entities
- GEO\_HIERARCHY -- documents the hierarchy of geographic objects (where it applies)

#### **Supportive DADS Dataset**

The DADS application requires access to supportive information such as:

- information about predefined products available for DADS access
- information about datasets available for ad-hoc query
- key word searches to predefined products and datasets
- subject area searches to predefined products and datasets
- user auditing information
- etc.

This physical database architecture follows its logical ERD model. The DADS team will load this data. Whenever possible, this data will be loaded from electronic sources, however, there may be the need to enter some data manually. In the next prototype, an Intranet application will be written against some of these tables to allow privileged Census users to create and post their own predefined products on the DADS system.

See Appendix B for a full listing of database tables for Decennial, Geography, and the Supportive DADS datasets.

#### SECTION 4

# AUDTING MODULE DESIGN Bureau of The Census Data Access and Dissemination System

#### **Prepared by DADS Back-End Team**

Author: Gail Wright Creation Date: July 19, 1996 Last Updated: July 19, 1996

Version:

#### **Audting Subsystem**

The purpose of this subsystem is to provide auditing information for a variety of uses including: database and application performance tuning, understanding how the DADS application is being used - what's popular and what's not, and more. The following is an Entity-Relationship Diagram of the data which will be collected by the auditing process.

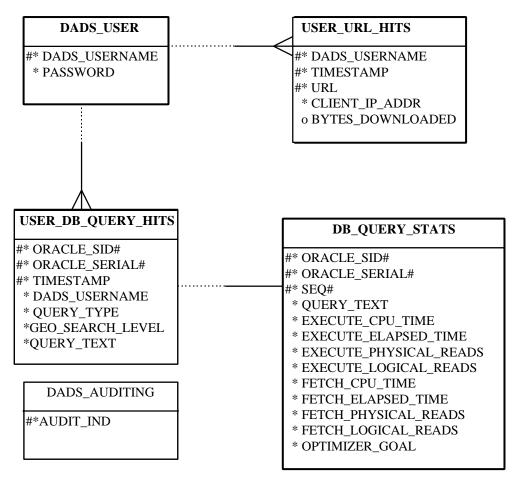


Figure 1: Auditing ERD

#### **General Overview**

For the first prototype, the main function of an auditing system will be data collection. Vendor software logging will be used to collect auditing information where possible. The Netscape Enterprise Server logging will be used to collect DADS application connection and usage information. The Oracle trace utility will be used to collect Oracle query statistics for queries executed through the DADS application. Other information about an Oracle query will be logged by the DADS application itself in the Oracle database. This will allow Oracle tracing to be linked back to a DADS user.

Later, the auditing system will be enhanced to add reporting and analysis capabilities. This will most likely occur during the beta testing period so that data collected during this period can be analyzed. One option is to load logging information into the Oracle database as shown in the Auditing ERD in Figure 1. In this way, auditing data can easily be related and queried for a variety of analysis and reports.

#### **General Functionality**

- Log information about a user's connection into an audit trail. (DADS1131)
- Log statistics to measure resource usage. (DADS1132)
- Log the user's major activities to understand how the user is using the system. (DADS1133)
- Log each user's ad-hoc query to understand what users are doing and to allow a user to re-access his/her query in the future. (DADS1134)
- Allow audit mode to be turned on and off. (DADS1135)

#### **DADS Login**

#### Overview

The DADS Login module is documented in the Front-End Modules Design document. Basically, the DADS development team will create and maintain usernames and passwords which are recognized by the Netscape Enterprise Server. Each beta test user will be given a username and password. Only this username/password combination will be accepted for connection into the DADS system. This will allow the DADS team to control the beta test, and to track the activity of individual beta testers in order to categorize types of users.

#### **Netscape Enterprise Server Logging**

#### Overview

Netscape Enterpriser Server Logging can be turned on or off. When turned on, it will log information into an ascii log file. This log file will include data for all attributes displayed in the USER\_URL\_HITS entity where:

#### DADS USERNAME

Is the authuser recognized by Netscape Enterprise Server which the DADS beta test user uses to log into the DADS application.

#### TIMESTAMP

Is the date and time a user accesses a particular URL. The timestamp is in the format: [day/month/year:hour:minute:second zone].

#### URL

Is the complete URL of an HTML page or frame that a user accesses. This URL can be used to understand what activities a user performs in the DADS system.

#### CLIENT\_IP\_ADDR

Is the IP address of the client machine running Netscape or a Web browser.

#### • BYTES\_DOWNLOADED

This is the number of bytes of a HTML objects which is downloaded to the user's Web browser. Not all URL hits include data for bytes downloaded due to the redirection status.

#### **Functionality**

- Log information about a user's connection into an audit trail. (DADS1131)
- Log the user's major activities to understand how the user is using the system. (DADS1133)
- Allow audit mode to be turned on and off. (DADS1135)

Note the following changes and/or clarification of the designed functionality compared with the Function Hierarchy documented in the DADS Analysis Report

#### **DADS1131**

• In the first prototype, Title 13 privilege is determined by which system is being accessed (e.g. the outside Internet machines or the inside Intranet machines). If a user is able to connect to DADS through the Intranet, then they are assumed to have Title 13 privileges.

#### **DADS1133**

• The extent of what non-database query activities a user performs is only be determined by the information provided in the URL in this prototype. For example, if a GET call is used, the parameters passed will be found in the URL; however, if a POST call is used the environmental variables passed will not be captured by the URL.

#### **DADS1135**

• There is no single "switch" to turn auditing on and off. Rather there are 3 switches: one for Netscape Enterprise Server, one for Oracle Trace, and one for the DADS Application.

#### **Oracle Trace**

#### Overview

The Oracle Trace facility can be turned on or off by setting an initialization parameter (SQL\_TRACE=TRUE). When turned on, it will log information into ascii trace files. This trace file will include data for all attributes displayed in the DB\_QUERY\_STATS entity except for SEQ# where:

#### ORACLE\_SID#

The Oracle assigned session identifier.

#### ORACLE\_SERIAL#

The Oracle assigned session serial number.

#### QUERY\_TEXT

The SQL statement of the ad-hoc query being executed. NOTE: Oracle Trace files also include recursive SQL statement which we may want to ignore.

#### • EXECUTE\_CPU\_TIME

The CPU time in seconds to execute the SQL statement (e.g. 01.22 would be 1 second and 22 hundredths of a second).

#### EXECUTE\_ELAPSED\_TIME

The elapsed time in seconds to execute the SQL statement.

#### EXECUTE\_PHYSICAL\_READS

The total number of data blocks physically read from the database files on physical disk which are needed to execute the given SQL statement.

#### EXECUTE\_LOGICAL\_READS

The total number of Oracle memory buffers retrieved which are needed to execute the given SQL statement.

#### • FETCH\_CPU\_TIME

The CPU time in seconds to fetch the records returned by the SQL query.

#### • FETCH\_ELAPSED\_TIME

The elapsed time in seconds to fetch the records returned by the SQL query.

#### FETCH\_PHYSICAL\_READS

The total number of data blocks physically read from the database files on physical disk which are needed to fetch the records returned by a SQL query.

#### FETCH\_LOGICAL\_READS

The total number of Oracle memory buffers retrieved which are needed to fetch the records returned by a SQL query.

#### OPTIMIZER\_GOAL

The goal used by the Oracle optimizer. For example, if set to RULE, then the rule-based optimizer was used. If set to CHOOSE, then Oracle chooses between the rule-based optimizer and the cost-based optimizer, such that it would choose the cost based optimizer only if the queried tables were "analyzed" and statistics were available to the cost based optimizer.

#### **Functionality**

- Log statistics to measure resource usage. (DADS1132)
- Log each user's ad-hoc query to understand what users are doing and to allow a user to re-access his/her query in the future. (DADS1134)
- Allow audit mode to be turned on and off. (DADS1135)

Note the following changes and/or clarification of the designed functionality compared with the Function Hierarchy documented in the DADS Analysis Report

#### **DADS1132**

Will measure resources used to execute a database query and fetch returned records, but
will not measure other resources used (such as overall system resources used on the
application and database servers). System performance will be measured using system
utilities and are not considered part of the auditing system in the first prototype.

#### **DADS1134**

• A user's ad-hoc query will be stored by the auditing system, however, the functionality to provide this query for future user access is not a part of the first prototype.

#### **DADS1135**

• There is no single "switch" to turn auditing on and off. Rather there are 3 switches: one for Netscape Enterprise Server, one for Oracle Trace, and one for the DADS Application

#### **Application Auditing**

#### Overview

Application Auditing is needed to provide the linkage between Netscape Enterprise Service auditing and Oracle trace auditing. An application module will need to be developed to insert an audit record into the Oracle database before an ad-hoc query is executed on behalf of a user. The audit table will be modelled after the USER DB QUERY HITS entity where:

#### • ORACLE\_SID#

The Oracle assigned session identifier. Queried from the V\$SESSION view.

#### ORACLE\_SERIAL#

The Oracle assigned session serial number. Queried from the V\$SESSION view.

#### TIMESTAMP

Is the date and time this record is inserted - indicating that the user's query is about to be executed. (The Oracle SYSDATE)

#### DADS\_USERNAME

Is the username of the DADS beta test user. A parameter passed in from the application.

#### QUERY TYPE

Extraction, tabulation, or summarization. A parameter passed in from the application.

#### GEO\_SEARCH\_LEVEL

The highest geographic level used in the query's search (e.g. US, state, county, etc.) A parameter passed in from the application.

#### QUERY\_TEXT

The SQL statement ad-hoc query about to be executed. A parameter passed in from the application.

#### **Functionality**

- Log each user's ad-hoc query to understand what users are doing and to allow a user to re-access his/her query in the future. (DADS1134)
- Allow audit mode to be turned on and off. (DADS1135)

#### **Detailed Description**

- Ad-hoc query created by the application on behalf of the user and is ready to be sent to the database server for processing.
- Query DADS\_AUDITING table to determine if auditing is turned on:

SELECT audit\_ind INTO v\_audit\_ind FROM dads\_auditing;

Suggestion: Store as a database function which returns v\_audit\_ind.

• If auditing is turned on, then insert a record into the USER\_DB\_QUERY\_HITS table:

Register the client user so that the user's SID and SERIAL# can be queried:

SELECT TO\_CHAR(client\_seq.nextval)
INTO v\_client\_seq
FROM dual;

Call the DBMS\_APPLICATION\_INFO stored procedure to supply identifying info about the client: DBMS\_APPLICATION\_INFO.SET\_CLIENT\_INFO(v\_client\_seq);

Get values for ORACLE\_SID#, ORACLE\_SERIAL#, and TIMESTAMP from the database:

 $SELECT\ sid,\ serial\#,\ TO\_CHAR(sysdate,\ 'dd/mm/yyyy:hh24:mi:ss')$ 

INTO v\_oracle\_sid#, v\_oracle\_serial#, v\_timestamp

FROM v\$session

WHERE client\_info = v\_client\_seq;

INSERT INTO USER\_DB\_QUERY\_HITS VALUES...

NOTE: Pass in the values for DADS USERNAME, QUERY TYPE,

GEO\_SEARCH\_LEVEL, and QUERY\_TEXT from the application.

Suggestion: Store as a database procedure so that the entire logic is executed in a single user session.

#### SECTION 5

## TECHNICAL ARCHITECTURE FOR PROTOTYPE #1

## Bureau of The Census Data Access and Dissemination System

Author: Neil Cotton Creation Date: July 24, 1996 Last Updated: July 24, 1996

Version:

#### **DADS Technical Architecture for Prototype #1**

#### **Break down of Computers by Function:**

#### Database Server:

This server must be the most powerful in order to serve up hundreds of Gigabytes of data to up to 50 concurrent users. For this prototype, we are using a large UNIX SMP(Symmetric Multi-Processor) server. This server is also equipped with the most RAM and hard disk space of all the servers. A switched Fast Ethernet connection (100 mbs) will be used to communicate with the other DADS servers and link directly to the Census Bureau's backbone.

#### • Application Server:

This will be our Web Server machine. Our users will "log" onto this machine and it will serve up the HTML pages. A large amount of RAM is important to handle many concurrent users on this system. It is also very important to have a fast connection between this machine and the Database server. Therefore, we are using a Fast Ethernet to insure quick data transfers.

#### • Mapping Server:

We decided to have a separate server to serve geographic data (Thematic maps and Navigation Maps) because we have seen these type of processes eat up application servers before. By having this separate from the Application server, we should be able to ensure more consistent performance for both servers. CPU speed and RAM are important for performance on this machine.

See Appendix C for more detail.

#### **Breakdown of Computer Systems by Project Stage:**

Development and Performance Testing:

For this stage, we are using three existing UNIX workstations, networked together with Fiber (FDDI) cabling.

#### • Test and Updates:

We need to set this system up as similar as possible to the production systems. This will be our test bed before we go production, and for fixes after the production cycle has begun.

#### • Internal Production:

As stated above, having a Database server efficient at server hundreds of Gigabytes of data, and a high speed network connecting all three servers to the backbone are the most important factors for a successful prototype.

#### • External Production:

Because this system will only serve a fraction of the data that the Internal Production system will serve, we can concentrate on insuring that we maintain a high level of security here.

Appendix A:	<b>DADS Functions to Modules</b>	Mapping Matrix.
Data Access and Dissemin		Annendiy A: DADS Functions to Modules Manning Matry

	DADS111	DADS112	DADS113	DADS121	DADS12211	DADS12212	DADS12213	DADS12214	DADS12216
Login	Χ	Χ							
Search					Χ		Χ	Χ	Χ
Geography Browser						Χ			
List Products and Reports									
View Metadata Information									
View Products and Reports									
Download Products and Report	ts								
Summary Tape File									
List Tables									
List Data Items									
Specify Calculated Data Items									
Tabulate									
Summarize									
Count and Compute									
Restrict Record Set									
View Outcome									
Download Outcome									

	DADS12217	<b>DADS1222</b>	<b>DADS1223</b>	DADS12231	DADS12232	DADS122331	DADS122332	DADS12234
Login								
Search	X	Χ	Χ			Χ		X
Geography Browser	Χ							
List Products and Reports				Χ	Χ		Χ	
View Metadata Information								
View Products and Reports								
Download Products and Report								
Summary Tape File				Χ	Χ		Χ	
List Tables								
List Data Items								
Specify Calculated Data Items								
Tabulate								
Summarize								
Count and Compute								
Restrict Record Set								
View Outcome								
Download Outcome								

	<b>DADS1231</b>	<b>DADS1232</b>	DADS1241	<b>DADS1242</b>	<b>DADS1243</b>	DADS1244	<b>DADS1245</b>	DADS131	DADS13211
Login									
Search									X
Geography Browser									
List Products and Reports									
View Metadata Information	Χ	Χ							Χ
View Products and Reports	Χ	Χ							
<b>Download Products and Report</b>	X	Χ	Χ	Χ	Χ	Χ	Χ		
Summary Tape File									
List Tables									
List Data Items									
Specify Calculated Data Items									
Tabulate									
Summarize									
Count and Compute									
Restrict Record Set									
View Outcome									
Download Outcome	Χ	Χ	Х	Х	Χ				

	DADS13212	DADS13213	DADS13215	DADS13216	DADS13217	DADS13218	DADS1324	DADS1331
Login								
Search	Χ	Χ	Χ	Χ	Χ	Χ		
Geography Browser		Χ				X		
List Products and Reports								
View Metadata Information					Χ			
View Products and Reports								
Download Products and Repor	1							
Summary Tape File								
List Tables							X	X
List Data Items								Χ
Specify Calculated Data Items								X
Tabulate								Χ
Summarize								X
Count and Compute								Χ
Restrict Record Set								X
View Outcome								
Download Outcome								

	DADS13311	DADS13312	DADS13313	<b>DADS1332</b>	DADS133211	DADS133212	DADS133213	DADS133214
Login								
Search								
Geography Browser								
List Products and Reports								
View Metadata Information								
View Products and Reports								
Download Products and Repor	1							
Summary Tape File								
List Tables	Χ	Χ			X			
List Data Items		Χ		X			X	
Specify Calculated Data Items				X			X	
Tabulate								
Summarize								
Count and Compute								
Restrict Record Set						X		
View Outcome								
Download Outcome			X					

	DADS133221	DADS133222	DADS133223	DADS133224	DADS133225	<b>DADS1333</b>	DADS134	DADS1351
Login								
Search								
Geography Browser								
List Products and Reports								
View Metadata Information								
View Products and Reports								
Download Products and Report								
Summary Tape File								
List Tables	X							
List Data Items				X				
Specify Calculated Data Items				X				
Tabulate			Χ					
Summarize			Χ					
Count and Compute				X				
Restrict Record Set		Χ						
View Outcome						Χ	X	
Download Outcome						X		Х

	<b>DADS1352</b>	DADS1353	DADS13531	DADS1354	<b>DADS1356</b>	DADS1362	<b>DADS1364</b>	<b>DADS1365</b>	DADS137
Login									
Search									
Geography Browser									
List Products and Reports									
View Metadata Information									
View Products and Reports									
Download Products and Report									
Summary Tape File									
List Tables									
List Data Items									
Specify Calculated Data Items									
Tabulate									
Summarize									
Count and Compute									
Restrict Record Set									
View Outcome						Χ	Χ		
Download Outcome		X	X	X	X				Χ

	DADS14	DADS16	DADS17	DADS18
Login				
Search				
Geography Browser				
List Products and Reports				
View Metadata Information				
View Products and Reports				
Download Products and Report				
Summary Tape File				
List Tables				
List Data Items				
Specify Calculated Data Items				
Tabulate				
Summarize				
Count and Compute				
Restrict Record Set				
View Outcome				
Download Outcome				

Note:			
Note:			

Appendix B: Database Tables for Decenr DADS datasets.	nial, Geography, and the Supportive
Data Access and Dissemination System	Appendix B: DADS Database Table

```
REM
REM This ORACLE7 command file was generated by Oracle Server Generator
REM Version 5.5.8.0.0 on 08-JUL-96
REM
REM For application DADS version 2 database DADS
REM
REM TABLE
REM
        A HOUSING100
         A_PERSONS100
REM
REM
        HOUSING100
REM
        PERSONS100
REM Modified by GW 7/10/96
REM File: crtab detail.sql
REM Purpose: This script serves as a template for creating 1990 Decennial
            100% detail tables. It takes 2 arguments:
REM
REM
REM
            &1: FIPS State Code
            &2: Tablespace Name
REM
REM
REM For example: sqlplus un/pwd @crtab_detail 25 data
REM This allows creating a set of tables for each state and placing each set
REM of tables in a given tablespace.
REM
REM End Modification 7/10/96
REM
REM
SPOOL crtab_detail_&1
PROMPT
PROMPT Creating Table A_HOUSING100
CREATE TABLE a_housing100_&1(
                                 CHAR(11)
                                                            NOT NULL,
house id
 aunits1
                                 CHAR(1)
                                                            NOT NULL,
arooms
                                 CHAR(1)
                                                            NOT NULL,
atenure
                                 CHAR(1)
                                                            NOT NULL,
aacreage
                                 CHAR(1)
                                                            NOT NULL,
acommuse
                                 CHAR(1)
                                                            NOT NULL,
avalue
                                 CHAR(1)
                                                            NOT NULL,
                                                            NOT NULL,
arent1
                                 CHAR(1)
ameals
                                 CHAR(1)
                                                            NOT NULL,
avacncy2
                                 CHAR(1)
                                                            NOT NULL,
avacncy3
                                 CHAR(1)
                                                            NOT NULL,
                                 CHAR(1)
                                                            NOT NULL
avacncy4
STORAGE (
  INITIAL
           134217728
 NEXT 134217728
 MINEXTENTS 1
 MAXEXTENTS 99
  PCTINCREASE 0
TABLESPACE &2
REM
PROMPT
PROMPT Creating Table A_PERSONS100
CREATE TABLE a_persons100_&1 (
pers_id
                                 CHAR(10)
                                                            NOT NULL,
```

```
REM
```

REM This ORACLE7 command file was generated by Oracle Server Generator

REM Version 5.5.8.0.0 on 08-JUL-96

REM

REM For application DADS version 2 database DADS

REM

REM TABLE

REM A\_HOUSING100

REM A\_PERSONS100

REM HOUSING100

REM PERSONS100

REM Modified by GW 7/10/96 REM File: crtab\_detail.sql

REM Purpose: This script serves as a template for creating 1990

Decennial

REM 100% detail tables. It takes 2 arguments:

REM

REM &1: FIPS State Code REM &2: Tablespace Name

REM

REM For example: sqlplus un/pwd @crtab\_detail 25 data

REM

REM This allows creating a set of tables for each state and placing each

REM of tables in a given tablespace.

REM

REM End Modification 7/10/96

REM

REM

SPOOL crtab\_detail\_&1

PROMPT

PROMPT Creating Table A\_HOUSING100

CREATE TABLE a\_housing100\_&1(

house_id	CHAR(11)	NOT NULL,
aunits1	CHAR(1)	NOT NULL,
arooms	CHAR(1)	NOT NULL,
atenure	CHAR(1)	NOT NULL,
aacreage	CHAR(1)	NOT NULL,

```
REM
    This ORACLE7 command file was generated by Oracle Server Generator
REM Version 5.5.8.2.0 on 06-AUG-96
REM For application DADS version 2 database DADS
REM
REM TABLE
REM
        SUMLEVEL 020
        SUMLEVEL 030
REM
        SUMLEVEL 040
        SUMLEVEL 050
REM
REM
        SUMLEVEL_060
REM
        SUMLEVEL_070
REM
        SUMLEVEL_080
REM
        SUMLEVEL_090
REM
        SUMLEVEL 140
REM
        SUMLEVEL 150
REM
        SUMLEVEL 155
REM
        SUMLEVEL 160
REM
        SUMLEVEL 170
REM
        SUMLEVEL 175
        SUMLEVEL 300
REM
        SUMLEVEL 301
REM
        SUMLEVEL 310
REM
REM
        SUMLEVEL_311
REM
        SUMLEVEL_319
REM
        SUMLEVEL_321
REM
        SUMLEVEL_329
REM
        SUMLEVEL 331
REM
        SUMLEVEL 340
REM
        SUMLEVEL 341
REM
        SUMLEVEL 400
        SUMLEVEL 410
REM
        SUMLEVEL 420
REM
        SUMLEVEL 430
REM
        SUMLEVEL 440
REM
REM
        SUMLEVEL_450
REM
        SUMLEVEL_460
REM
        SUMLEVEL_501
REM
         SUMLEVEL_510
REM
        SUMLEVEL_530
REM
        SUMLEVEL_540
REM
         SUMLEVEL 800
REM
         SUMLEVEL 820
REM Modified by GW 8/6/96
REM File: crtab geo.sql
REM Purpose: This script creates the 1990 Geography tables. Each table
             represents a different STF (Summary Tape File) summary
REM
             level geography. The source geography for loading these
             tables will come from the 1990 STF geography data.
REM
REM
REM
             This script has one argument:
               &1: Tablespace Name
REM For example: sqlplus un/pwd @crtab_geo geodata
REM This allows for creating all geographic tables in a separate
tablespace.
REM End Modification 8/6/96
```

```
REM
REM This ORACLE7 command file was generated by Oracle Server Generator
REM Version 5.5.8.0.0 on 10-JUL-96
REM For application DADS version 2 database DADS
REM
REM TABLE
      DADS PRODUCTS
REM
       DB TABLES
REM
REM
       DB TABLE COLUMNS
       FILES
REM
REM
      GEO_HIERARCHY
REM
      GEO_OBJECTS
REM
      KEYWORDS
REM
      KEYWORD ASSIGNMENTS
REM
      KEYWORD_SYNONYM_ASSIGNMENTS
REM
      PREDEFINED_TYPES
REM
      PRODUCT DISTRIBUTION FORMATS
      PRODUCT FORMATS
REM
      RELATED ORGS
REM
REM
       SERVERS
REM
       SUBJECT AREAS
       SUBJECT ASSIGNMENTS
REM
       VALID_VALUES
REM
REM Modified by GW 7/10/96
REM File: crtab_metadata.sql
REM Purpose: This script creates the supportive tables needed by the
            application. It takes one argument:
REM
REM
               &1: Tablespace Name
REM For example: sqlplus un/pwd @crtab metadata metadata
REM This allows for creating all supportive tables in a separate
tablespace.
REM
REM End Modification 7/10/96
REM Modified 8/7/96 by GW - Dropped table URLS and modified table
                           PRODUCT_DISTRIBUTION_FORMATS by adding a url
REM
column.
REM
SPOOL crtab metadata
PROMPT
PROMPT Creating Table DADS_PRODUCTS
CREATE TABLE dads_products(
                                NUMBER(8,0)
                                                          NOT NULL,
 predefined_ind
                                CHAR(1)
                                                          NOT NULL,
 predefined_type
                               CHAR (20)
                                                          NULL,
 short_name
                               VARCHAR2(15)
                                                          NOT NULL,
 long name
                               VARCHAR2(60)
                                                         NULL,
 long descr
                               VARCHAR2(255)
                                                         NULL,
 version
                               VARCHAR2(10)
                                                         NOT NULL,
 content accuracy beg date
                               DATE
                                                         NOT NULL,
 content accuracy end date
                               DATE
                                                         NULL,
 title_13_security
                                                          NOT NULL,
                               CHAR(1)
 availability
                               CHAR(1)
                                                          NOT NULL,
```

```
availability_msg
                                 VARCHAR2(60)
                                                             NULL,
 sponsor_org_id
                                 NUMBER
                                                             NULL,
 content_provider_org_id
                                 NUMBER
                                                             NULL,
all_geo_levels_ind
                                 CHAR(1)
                                                             NULL,
usa level ind
                                 CHAR(1)
                                                             NULL,
region ind
                                 CHAR(1)
                                                             NULL,
division ind
                                 CHAR(1)
                                                             NULL,
state ind
                                 CHAR(1)
                                                             NULL,
county ind
                                 CHAR(1)
                                                             NULL,
msa ind
                                 CHAR(1)
                                                             NULL,
cmsa ind
                                 CHAR(1)
                                                             NULL,
place_ind
                                 CHAR(1)
                                                             NULL,
                                 CHAR(1)
region
                                                             NULL,
division
                                 CHAR(1)
                                                             NULL,
state
                                 CHAR(2)
                                                             NULL,
county
                                 CHAR(3)
                                                             NULL,
msa
                                 CHAR (4)
                                                             NULL,
                                 CHAR(2)
                                                             NULL,
cmsa
place
                                 CHAR(4)
                                                             NULL,
creation date
                                 DATE
                                                             NOT NULL,
who created
                                 VARCHAR2(70)
                                                             NOT NULL,
modified date
                                 DATE
                                                             NULL,
who_modified
                                 VARCHAR2(70)
                                                             NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT
       131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
  )
;
REM
PROMPT Creating Table DB_TABLES
CREATE TABLE db_tables(
table_name
                                 VARCHAR2(30)
                                                             NOT NULL,
description
                                 VARCHAR2 (255)
                                                             NULL,
product_id
                                 NUMBER (8)
                                                             NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table DB TABLE COLUMNS
CREATE TABLE db table columns(
table name
                                 VARCHAR2(30)
                                                             NOT NULL,
column name
                                 VARCHAR2(8)
                                                             NOT NULL,
short desc
                                 VARCHAR2(40)
                                                             NOT NULL,
show hide ind
                                 CHAR(1)
                                                             NULL,
```

```
CHAR(1)
                                                        NOT NULL,
 range_ind
                               VARCHAR2(10)
                                                        NOT NULL,
 datatype
 col_length
                               NUMBER (8)
                                                        NULL,
                            NUMBER(2)
VARCHAR2(255)
VARCHAR2(8)
VARCHAR2(2000)
 decimal_places
                                                      NULL,
                                                      NULL,
 description
 variable name
                                                      NULL,
 census_question
                                                      NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
REM
PROMPT Creating Table FILES
CREATE TABLE files(
                                                      NOT NULL,
NOT NULL,
NOT NULL,
file id
                               NUMBER(10,0)
filename
                               VARCHAR2(70)
server_id
                             NUMBER (8)
product_id
                             NUMBER (8)
                            VARCHAR2(15)
product_format_type
                                                       NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table GEO_HIERARCHY
CREATE TABLE geo_hierarchy(
NULL,
                                                        NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table GEO OBJECTS
CREATE TABLE geo_objects(
geo object
                              VARCHAR2(8)
                                                      NOT NULL,
 object_name
                               VARCHAR2(45)
                                                        NOT NULL,
```

```
description
                               VARCHAR2(255)
                                                          NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
REM
PROMPT Creating Table KEYWORDS
CREATE TABLE keywords(
keyword
                               VARCHAR2(30)
                                                         NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table KEYWORD ASSIGNMENTS
CREATE TABLE keyword assignments(
                                                      NOT NULL,
assignment id
                               NUMBER(10,0)
keyword
                               VARCHAR2(30)
                                                        NOT NULL,
product_id
                               NUMBER(8,0)
                                                         NULL,
column_name
                               VARCHAR2(30)
                                                         NULL,
value_id
                               NUMBER
                                                          NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
PROMPT Creating Table KEYWORD_SYNONYM_ASSIGNMENTS
CREATE TABLE keyword_synonym_assignments(
                               VARCHAR2(30)
                                                         NOT NULL,
keyword
                               VARCHAR2(30)
                                                         NOT NULL
synonym_name
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
```

```
MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
  )
;
REM
PROMPT
PROMPT Creating Table PREDEFINED TYPES
CREATE TABLE predefined_types(
                                VARCHAR2(20)
                                                           NOT NULL,
predefined_type
                                VARCHAR2(255)
                                                           NULL
description
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table PRODUCT_DISTRIBUTION_FORMATS
CREATE TABLE product_distribution_formats(
product_id
                                NUMBER (8,0)
                                                           NOT NULL,
product_format_type
                                VARCHAR2(15)
                                                          NOT NULL,
 orderable
                                CHAR(1)
                                                          NOT NULL,
 available
                                CHAR(1)
                                                          NOT NULL,
 version
                                VARCHAR2(10)
                                                          NOT NULL,
 byte size
                                NUMBER
                                                          NULL,
 page_size
                               NUMBER
                                                          NULL,
 description
                               VARCHAR2 (255)
                                                          NULL,
 sw_tips
                               VARCHAR2 (255)
                                                          NULL,
url
                                VARCHAR2(78)
                                                           NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table PRODUCT_FORMATS
CREATE TABLE product_formats(
                                VARCHAR2(15)
                                                           NOT NULL,
 format_type
description
                                VARCHAR2(80)
                                                           NULL,
                                VARCHAR2 (255)
                                                           NULL
sw_tips
TABLESPACE &1
STORAGE (
 INITIAL 8192
```

```
NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table RELATED ORGS
CREATE TABLE related_orgs(
                               NUMBER
                                                         NOT NULL,
org_id
org_name
                               VARCHAR2(50)
                                                         NOT NULL,
contact
                               VARCHAR2(50)
                                                         NULL,
 contact_phone
                               VARCHAR2(12)
                                                         NULL,
 contact_internet_addr
                               VARCHAR2(70)
                                                         NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
PROMPT Creating Table SERVERS
CREATE TABLE servers(
server id
                               NUMBER(8,0)
                                                        NOT NULL,
ip address
                               VARCHAR2(20)
                                                        NOT NULL,
server name
                               VARCHAR2(40)
                                                        NOT NULL,
httpd_port
                               NUMBER
                                                         NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
PROMPT Creating Table SUBJECT_AREAS
CREATE TABLE subject_areas(
subject
                               VARCHAR2(30)
                                                         NOT NULL,
description
                               VARCHAR2(80)
                                                          NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
```

```
PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUBJECT ASSIGNMENTS
CREATE TABLE subject_assignments(
product id
                               NUMBER(8,0)
                                                       NOT NULL,
subject
                               VARCHAR2(30)
                                                        NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table VALID_VALUES
CREATE TABLE valid_values(
                                                      NOT NULL,
value_id
                               NUMBER(8,0)
table_name
                               VARCHAR2(30)
 column_name
                               VARCHAR2(8)
                                                       NOT NULL,
 low value
                              VARCHAR2(10)
                                                       NOT NULL,
high value
                             VARCHAR2(10)
                                                       NULL,
meaning
                             VARCHAR2(50)
                                                       NULL,
 comments
                              VARCHAR2 (255)
                                                        NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
SPOOL off
```

```
REM
REM
SPOOL crtab_geo
PROMPT Creating Table SUMLEVEL 020
CREATE TABLE sumlevel 020(
                                NUMBER(8)
                                                          NOT NULL,
region
                                CHAR(1)
                                                          NOT NULL,
name
                                VARCHAR2(80)
                                                          NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_030
CREATE TABLE sumlevel_030(
                                NUMBER (8)
                                                          NOT NULL,
geoid
region
                                CHAR(1)
                                                          NOT NULL,
division
                                CHAR(1)
                                                          NOT NULL,
name
                                VARCHAR2(80)
                                                          NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL 040
CREATE TABLE sumlevel 040(
geoid
                                NUMBER(8)
                                                          NOT NULL,
region
                                CHAR(1)
                                                          NOT NULL,
division
                                CHAR(1)
                                                         NOT NULL,
censtate
                                CHAR(2)
                                                          NOT NULL,
                                CHAR(2)
                                                         NOT NULL,
state
                                VARCHAR2(80)
                                                          NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
```

```
REM
PROMPT
PROMPT Creating Table SUMLEVEL_050
CREATE TABLE sumlevel_050(
                                NUMBER(8)
                                                         NOT NULL,
geoid
                                CHAR(2)
                                                         NOT NULL,
censtate
                                CHAR(2)
                                                          NOT NULL,
state
                                CHAR(3)
                                                         NOT NULL,
county
name
                                VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL 060
CREATE TABLE sumlevel_060(
 geoid
                                NUMBER (8)
                                                           NOT NULL,
                                CHAR(2)
                                                           NOT NULL,
 censtate
state
                                CHAR(2)
                                                           NOT NULL,
 county
                                CHAR(3)
                                                           NOT NULL,
 mcd
                                CHAR(3)
                                                           NOT NULL,
                                CHAR(5)
                                                           NOT NULL,
 cousubfp
name
                                VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL 070
CREATE TABLE sumlevel 070(
geoid
                                NUMBER (8)
                                                           NOT NULL,
 censtate
                                CHAR(2)
                                                           NOT NULL,
                                CHAR(2)
                                                           NOT NULL,
 state
                                CHAR(3)
                                                           NOT NULL,
 county
                                CHAR(3)
                                                           NOT NULL,
 mcd
 cousubfp
                                CHAR(5)
                                                           NOT NULL,
place
                                CHAR(4)
                                                           NOT NULL,
placefp
                                CHAR(5)
                                                           NOT NULL,
                                VARCHAR2(80)
                                                           NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
```

```
MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
  )
;
REM
PROMPT Creating Table SUMLEVEL 080
CREATE TABLE sumlevel 080(
                                 NUMBER (8)
                                                            NOT NULL,
                                 CHAR(2)
                                                            NOT NULL,
censtate
state
                                 CHAR(2)
                                                            NOT NULL,
                                 CHAR(3)
                                                            NOT NULL,
county
mcd
                                 CHAR(3)
                                                            NOT NULL,
 cousubfp
                                 CHAR(5)
                                                            NOT NULL,
place
                                CHAR(4)
                                                           NOT NULL,
placefp
                                CHAR(5)
                                                           NOT NULL,
tract
                                CHAR(6)
                                                           NOT NULL,
name
                                 VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 4194304
 NEXT 4194304
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL 090
CREATE TABLE sumlevel 090(
geoid
                                 NUMBER (8)
                                                            NOT NULL,
censtate
                                 CHAR(2)
                                                            NOT NULL,
                                 CHAR(2)
                                                            NOT NULL,
state
                                 CHAR(3)
                                                            NOT NULL,
county
                                 CHAR(3)
                                                            NOT NULL,
mcd
 cousubfp
                                 CHAR(5)
                                                            NOT NULL,
place
                                 CHAR(4)
                                                            NOT NULL,
placefp
                                CHAR(5)
                                                           NOT NULL,
tract
                                CHAR(6)
                                                           NOT NULL,
blkgroup
                                CHAR(1)
                                                           NOT NULL,
name
                                 VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 4194304
 NEXT 4194304
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL 140
CREATE TABLE sumlevel_140(
```

```
geoid
                               NUMBER (8)
                                                         NOT NULL,
 censtate
                               CHAR(2)
                                                         NOT NULL,
 state
                               CHAR(2)
                                                         NOT NULL,
 county
                               CHAR(3)
                                                        NOT NULL,
 tract
                               CHAR(6)
                                                       NOT NULL,
name
                               VARCHAR2(80)
                                                         NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 4194304
 NEXT 4194304
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL 150
CREATE TABLE sumlevel 150(
                               NUMBER(8)
                                                       NOT NULL,
geoid
censtate
                               CHAR(2)
                                                       NOT NULL,
state
                               CHAR(2)
                                                       NOT NULL,
                               CHAR(3)
                                                       NOT NULL,
county
 tract
                              CHAR(6)
                                                       NOT NULL,
                              CHAR(1)
                                                       NOT NULL,
blkgroup
                              VARCHAR2(80)
                                                         NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 4194304
 NEXT 4194304
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_155
CREATE TABLE sumlevel 155(
                               NUMBER (8)
                                                       NOT NULL,
geoid
censtate
                               CHAR(2)
                                                       NOT NULL,
state
                               CHAR(2)
                                                       NOT NULL,
place
                               CHAR (4)
                                                       NOT NULL,
placefp
                              CHAR(5)
                                                       NOT NULL,
county
                              CHAR(3)
                                                       NOT NULL,
name
                              VARCHAR2(80)
                                                        NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
```

```
REM
PROMPT Creating Table SUMLEVEL_160
CREATE TABLE sumlevel_160(
                                NUMBER (8)
                                                           NOT NULL,
                                CHAR(2)
                                                           NOT NULL,
censtate
state
                                CHAR(2)
                                                          NOT NULL,
place
                                CHAR(4)
                                                          NOT NULL,
placefp
                                CHAR(5)
                                                         NOT NULL,
name
                                VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL 170
CREATE TABLE sumlevel_170(
geoid
                                NUMBER (8)
                                                           NOT NULL,
                                CHAR(2)
                                                           NOT NULL,
censtate
state
                                CHAR(2)
                                                           NOT NULL,
CC
                                CHAR(1)
                                                           NOT NULL,
concitfp
                                CHAR(5)
                                                           NOT NULL,
                                VARCHAR2(80)
                                                           NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL 175
CREATE TABLE sumlevel 175(
                                NUMBER (8)
                                                           NOT NULL,
geoid
censtate
                                CHAR(2)
                                                           NOT NULL,
state
                                CHAR(2)
                                                          NOT NULL,
                                CHAR(1)
                                                          NOT NULL,
CC
                                CHAR(5)
                                                           NOT NULL,
 concitfp
                                CHAR(3)
                                                           NOT NULL,
county
name
                                VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
```

```
)
REM
PROMPT
PROMPT Creating Table SUMLEVEL 300
CREATE TABLE sumlevel 300(
                                NUMBER(8)
                                                          NOT NULL,
geoid
cmsa
                                CHAR(2)
                                                          NOT NULL,
msacmsa
                                CHAR (4)
                                                          NOT NULL,
name
                                VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_301
CREATE TABLE sumlevel_301(
                                NUMBER (8)
                                                           NOT NULL,
geoid
cmsa
                                CHAR(2)
                                                           NOT NULL,
pmsa
                                CHAR (4)
                                                           NOT NULL,
name
                                VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL 310
CREATE TABLE sumlevel 310(
                                NUMBER(8)
                                                           NOT NULL,
geoid
cmsa
                                CHAR(2)
                                                           NOT NULL,
msacmsa
                                CHAR(4)
                                                           NOT NULL,
censtate
                                CHAR(2)
                                                           NOT NULL,
                                CHAR(2)
                                                           NOT NULL,
state
                                VARCHAR2(80)
                                                           NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
```

```
REM
PROMPT
PROMPT Creating Table SUMLEVEL_311
CREATE TABLE sumlevel_311(
                                 NUMBER(8)
                                                           NOT NULL,
geoid
                                 CHAR(2)
                                                           NOT NULL,
 cmsa
                                 CHAR(4)
                                                          NOT NULL,
pmsa
                                 CHAR(2)
                                                          NOT NULL,
censtate
state
                                CHAR(2)
                                                          NOT NULL,
                                VARCHAR2(80)
                                                           NOT NULL
name
TABLESPACE &1
STORAGE (
  INITIAL 8192
  NEXT 8192
  MINEXTENTS 1
  MAXEXTENTS 99
  PCTINCREASE 0
  )
REM
PROMPT Creating Table SUMLEVEL_319
CREATE TABLE sumlevel_319(
                                 NUMBER (8)
                                                           NOT NULL,
geoid
                                 CHAR(2)
                                                           NOT NULL,
censtate
state
                                 CHAR(2)
                                                           NOT NULL,
                                CHAR(2)
                                                           NOT NULL,
 cmsa
                                CHAR(4)
                                                           NOT NULL,
msacmsa
name
                                VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
  INITIAL 8192
  NEXT 32768
  MINEXTENTS 1
  MAXEXTENTS 99
  PCTINCREASE 0
  )
;
REM
PROMPT Creating Table SUMLEVEL 321
CREATE TABLE sumlevel 321(
geoid
                                 NUMBER (8)
                                                           NOT NULL,
 censtate
                                 CHAR(2)
                                                           NOT NULL,
                                 CHAR(2)
                                                           NOT NULL,
 state
                                 CHAR(2)
                                                           NOT NULL,
 cmsa
                                CHAR(4)
                                                           NOT NULL,
pmsa
name
                                VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
  INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
  PCTINCREASE 0
```

```
)
REM
PROMPT
PROMPT Creating Table SUMLEVEL 329
CREATE TABLE sumlevel 329(
                                 NUMBER(8)
                                                            NOT NULL,
geoid
censtate
                                 CHAR(2)
                                                            NOT NULL,
state
                                 CHAR(2)
                                                            NOT NULL,
msacmsa
                                 CHAR(4)
                                                            NOT NULL,
                                 CHAR(3)
                                                            NOT NULL,
county
                                 VARCHAR2(80)
                                                            NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT
       32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_331
CREATE TABLE sumlevel_331(
geoid
                                 NUMBER (8)
                                                            NOT NULL,
censtate
                                 CHAR(2)
                                                            NOT NULL,
                                 CHAR(2)
                                                            NOT NULL,
state
cmsa
                                 CHAR(2)
                                                            NOT NULL,
                                 CHAR(4)
                                                            NOT NULL,
pmsa
                                 CHAR(3)
                                                            NOT NULL,
county
                                 VARCHAR2(80)
                                                            NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_340
CREATE TABLE sumlevel_340(
                                 NUMBER (8)
                                                            NOT NULL,
geoid
                                 CHAR(2)
                                                            NOT NULL,
censtate
state
                                 CHAR(2)
                                                            NOT NULL,
msacmsa
                                 CHAR(4)
                                                            NOT NULL,
place
                                 CHAR (4)
                                                            NOT NULL,
placefp
                                 CHAR(5)
                                                            NOT NULL,
                                 VARCHAR2(80)
                                                            NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 8192
```

```
NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL 341
CREATE TABLE sumlevel_341(
                                 NUMBER (8)
                                                            NOT NULL,
geoid
                                 CHAR(2)
                                                            NOT NULL,
censtate
                                 CHAR(2)
                                                            NOT NULL,
state
cmsa
                                 CHAR(2)
                                                            NOT NULL,
pmsa
                                 CHAR (4)
                                                            NOT NULL,
place
                                 CHAR(4)
                                                            NOT NULL,
placefp
                                 CHAR(5)
                                                           NOT NULL,
name
                                 VARCHAR2(80)
                                                            NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL 400
CREATE TABLE sumlevel 400(
                                 NUMBER (8)
                                                           NOT NULL,
geoid
ua
                                 CHAR(4)
                                                           NOT NULL,
name
                                 VARCHAR2(80)
                                                            NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT
       32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL_410
CREATE TABLE sumlevel_410(
geoid
                                 NUMBER (8)
                                                            NOT NULL,
ua
                                 CHAR(4)
                                                            NOT NULL,
censtate
                                 CHAR(2)
                                                            NOT NULL,
                                 CHAR(2)
                                                           NOT NULL,
state
                                 VARCHAR2(80)
                                                            NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 8192
```

```
NEXT 32768
  MINEXTENTS 1
  MAXEXTENTS 99
  PCTINCREASE 0
  )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL 420
CREATE TABLE sumlevel 420(
                                 NUMBER (8)
                                                            NOT NULL,
 geoid
                                 CHAR(2)
                                                           NOT NULL,
censtate
state
                                 CHAR(2)
                                                           NOT NULL,
ua
                                 CHAR(4)
                                                           NOT NULL,
name
                                 VARCHAR2(80)
                                                            NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_430
CREATE TABLE sumlevel 430(
geoid
                                 NUMBER(8)
                                                           NOT NULL,
censtate
                                 CHAR(2)
                                                           NOT NULL,
state
                                 CHAR(2)
                                                           NOT NULL,
                                 CHAR (4)
                                                           NOT NULL,
 ua
                                 CHAR(3)
                                                           NOT NULL,
 county
name
                                 VARCHAR2(80)
                                                            NOT NULL
TABLESPACE &1
STORAGE (
  INITIAL 8192
  NEXT
       32768
  MINEXTENTS 1
 MAXEXTENTS 99
  PCTINCREASE 0
  )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL_440
CREATE TABLE sumlevel_440(
 geoid
                                 NUMBER (8)
                                                            NOT NULL,
censtate
                                 CHAR(2)
                                                            NOT NULL,
state
                                 CHAR(2)
                                                            NOT NULL,
                                 CHAR(4)
                                                            NOT NULL,
 ua
                                 CHAR(3)
                                                           NOT NULL,
 county
mcd
                                 CHAR(3)
                                                           NOT NULL,
 cousubfp
                                CHAR(5)
                                                           NOT NULL,
name
                                VARCHAR2(80)
                                                           NOT NULL
```

```
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL_450
CREATE TABLE sumlevel_450(
geoid
                                NUMBER (8)
                                                           NOT NULL,
censtate
                                CHAR(2)
                                                           NOT NULL,
                                CHAR(2)
                                                           NOT NULL,
state
                                CHAR(4)
                                                          NOT NULL,
ua
                                CHAR(3)
                                                          NOT NULL,
county
mcd
                                CHAR(3)
                                                          NOT NULL,
cousubfp
                                CHAR (5)
                                                          NOT NULL,
place
                                CHAR (4)
                                                          NOT NULL,
placefp
                                CHAR (5)
                                                          NOT NULL,
                                VARCHAR2(80)
                                                           NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_460
CREATE TABLE sumlevel_460(
geoid
                                NUMBER (8)
                                                           NOT NULL,
censtate
                                CHAR(2)
                                                           NOT NULL,
state
                                CHAR(2)
                                                          NOT NULL,
                                CHAR(4)
                                                          NOT NULL,
ua
place
                                CHAR(4)
                                                          NOT NULL,
placefp
                                CHAR(5)
                                                          NOT NULL,
name
                                VARCHAR2(80)
                                                           NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_501
```

```
CREATE TABLE sumlevel_501(
                                                      NOT NULL,
                               NUMBER (8)
 geoid
 censtate
                               CHAR(2)
 state
                               CHAR(2)
                                                       NOT NULL,
 cd103
                               CHAR(2)
                                                       NOT NULL,
name
                               VARCHAR2(80)
                                                         NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 32768
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL 510
CREATE TABLE sumlevel 510(
                                                       NOT NULL,
                               NUMBER (8)
geoid
censtate
                               CHAR(2)
                                                       NOT NULL,
state
                               CHAR(2)
                                                       NOT NULL,
 cd103
                               CHAR(2)
                                                       NOT NULL,
county
                               CHAR(3)
                                                       NOT NULL,
                               VARCHAR2(80)
                                                        NOT NULL
name
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_530
CREATE TABLE sumlevel_530(
geoid
                               NUMBER(8)
                                                       NOT NULL,
                               CHAR(2)
                                                       NOT NULL,
censtate
state
                               CHAR(2)
                                                       NOT NULL,
cd103
                               CHAR(2)
                                                       NOT NULL,
place
                               CHAR (4)
                                                       NOT NULL,
placefp
                              CHAR (5)
                                                       NOT NULL,
name
                              VARCHAR2(80)
                                                        NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
REM
```

```
PROMPT
PROMPT Creating Table SUMLEVEL_540
CREATE TABLE sumlevel_540(
 geoid
                               NUMBER (8)
                                                       NOT NULL,
                                                         NOT NULL,
 censtate
                               CHAR(2)
state
                               CHAR(2)
                                                       NOT NULL,
 cd103
                               CHAR(2)
                                                       NOT NULL,
                               CHAR(1)
                                                       NOT NULL,
 CC
 concitfp
                              CHAR (5)
                                                       NOT NULL,
name
                              VARCHAR2(80)
                                                        NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 8192
 NEXT 8192
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT
PROMPT Creating Table SUMLEVEL_800
CREATE TABLE sumlevel_800(
                                                       NOT NULL,
geoid
                               NUMBER(8)
zipcode
                               CHAR(5)
                                                        NOT NULL,
name
                               VARCHAR2(80)
                                                         NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 131072
 NEXT 131072
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
;
REM
PROMPT Creating Table SUMLEVEL_820
CREATE TABLE sumlevel 820(
                                                      NOT NULL,
geoid
                               NUMBER (8)
zipcode
                               CHAR(5)
county
                               CHAR(3)
                                                       NOT NULL,
name
                               VARCHAR2(80)
                                                         NOT NULL
TABLESPACE &1
STORAGE (
 INITIAL 2097152
 NEXT 2097152
 PCTINCREASE 0
 )
spool off
```

```
acommuse
                                  CHAR(1)
                                                             NOT NULL,
avalue
                                  CHAR(1)
                                                             NOT NULL,
arent1
                                  CHAR(1)
                                                             NOT NULL,
ameals
                                  CHAR(1)
                                                             NOT NULL,
avacncy2
                                  CHAR(1)
                                                             NOT NULL,
avacncy3
                                  CHAR(1)
                                                             NOT NULL,
avacncy4
                                  CHAR(1)
                                                             NOT NULL
STORAGE (
 INITIAL 134217728
 NEXT 134217728
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
  )
TABLESPACE &2
;
REM
PROMPT
PROMPT Creating Table A_PERSONS100
CREATE TABLE a_persons100_&1 (
pers_id
                                  CHAR(10)
                                                             NOT NULL,
arelat1
                                  CHAR(1)
                                                             NOT NULL,
asex
                                  CHAR(1)
                                                             NOT NULL,
arace
                                  CHAR(1)
                                                             NOT NULL,
aage
                                  CHAR(1)
                                                             NOT NULL,
amarital
                                  CHAR(1)
                                                             NOT NULL,
ahispan
                                  CHAR(1)
                                                             NOT NULL
STORAGE (
```

```
INITIAL
            134217728
 NEXT
         134217728
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
  )
TABLESPACE &2
REM
PROMPT
PROMPT Creating Table HOUSING100
CREATE TABLE housing100_&1 (
house_id
                                  CHAR(11)
                                                               NOT NULL,
houstype
                                  CHAR(1)
                                                               NOT NULL,
units1
                                  CHAR(2)
                                                               NOT NULL,
rooms
                                  NUMBER(1)
                                                               NOT NULL,
 tenure
                                  CHAR(1)
                                                               NOT NULL,
acreage
                                  CHAR(1)
                                                               NOT NULL,
commuse
                                  CHAR(1)
                                                               NOT NULL,
value
                                  CHAR(2)
                                                               NOT NULL,
rent1
                                  CHAR(2)
                                                               NOT NULL,
meals
                                  CHAR(1)
                                                               NOT NULL,
vacancy1
                                  CHAR(1)
                                                               NOT NULL,
vacancy2
                                  CHAR(1)
                                                               NOT NULL,
vacancy3
                                  CHAR(1)
                                                               NOT NULL,
vacancy4
                                  CHAR(1)
                                                               NOT NULL,
persons
                                  NUMBER (5)
                                                               NOT NULL,
perssrc
                                  CHAR(1)
                                                               NOT NULL,
 formtype
                                  CHAR(1)
                                                               NOT NULL,
                                                               NOT NULL,
hsub
                                  CHAR(1)
```

hpsub	CHAR(1)	NOT	NULL,
hsda	CHAR(1)	NOT	NULL,
gqtype	CHAR(3)	NOT	NULL,
gqtypesc	CHAR(1)	NOT	NULL,
gqinst	CHAR(1)	NOT	NULL,
gqinstsc	CHAR(1)	NOT	NULL,
unit_id	CHAR(7)	NOT	NULL,
state	CHAR(2)	NOT	NULL,
county	CHAR(3)	NOT	NULL,
tract	CHAR(6)	NOT	NULL,
blkgroup	CHAR(1)	NOT	NULL,
block	CHAR(2)	NOT	NULL,
zip	CHAR(5)	NOT	NULL,
msa	CHAR(4)	NOT	NULL,
cmsa	CHAR(2)	NOT	NULL,
place	CHAR(4)	NOT	NULL,
mcd	CHAR(3)	NOT	NULL,
cd101	CHAR(2)	NOT	NULL,
cd103	CHAR(2)	NOT	NULL,
anrc	CHAR(2)	NOT	NULL,
ua	CHAR(4)	NOT	NULL,
ur	CHAR(1)	NOT	NULL,
air	CHAR(4)	NOT	NULL,
geo_id	CHAR(11)	NOT	NULL
)			
STORAGE (			
INITIAL 134217728			
NEXT 134217728			
MINEXTENTS 1			
MAXEXTENTS 99			

```
PCTINCREASE 0
  )
TABLESPACE &2
REM
PROMPT
PROMPT Creating Table PERSONS100
CREATE TABLE persons100_&1 (
pers_id
                                   CHAR(10)
                                                                NOT NULL,
house_id
                                   CHAR(11)
                                                                NOT NULL,
 relat1
                                   CHAR(2)
                                                                NOT NULL,
 sex
                                   CHAR(1)
                                                                NOT NULL,
 race
                                   CHAR(3)
                                                                NOT NULL,
 age
                                   NUMBER(3)
                                                                NOT NULL,
marital
                                   CHAR(1)
                                                                NOT NULL,
hispanic
                                   CHAR(1)
                                                                NOT NULL,
 state
                                   CHAR(2)
                                                                NOT NULL,
 county
                                   CHAR(3)
                                                                NOT NULL,
 tract
                                   CHAR(6)
                                                                NOT NULL,
blkgroup
                                   CHAR(1)
                                                                NOT NULL,
block
                                   CHAR(2)
                                                                NOT NULL,
 zip
                                   CHAR (5)
                                                                NOT NULL,
 msa
                                   CHAR (4)
                                                                NOT NULL,
 cmsa
                                   CHAR(2)
                                                                NOT NULL,
place
                                   CHAR (4)
                                                                NOT NULL,
mcd
                                   CHAR(3)
                                                                NOT NULL,
 cd101
                                   CHAR(2)
                                                                NOT NULL,
 cd103
                                   CHAR(2)
                                                                NOT NULL,
 anrc
                                   CHAR(2)
                                                                NOT NULL,
                                   CHAR (4)
                                                                NOT NULL,
 ua
```

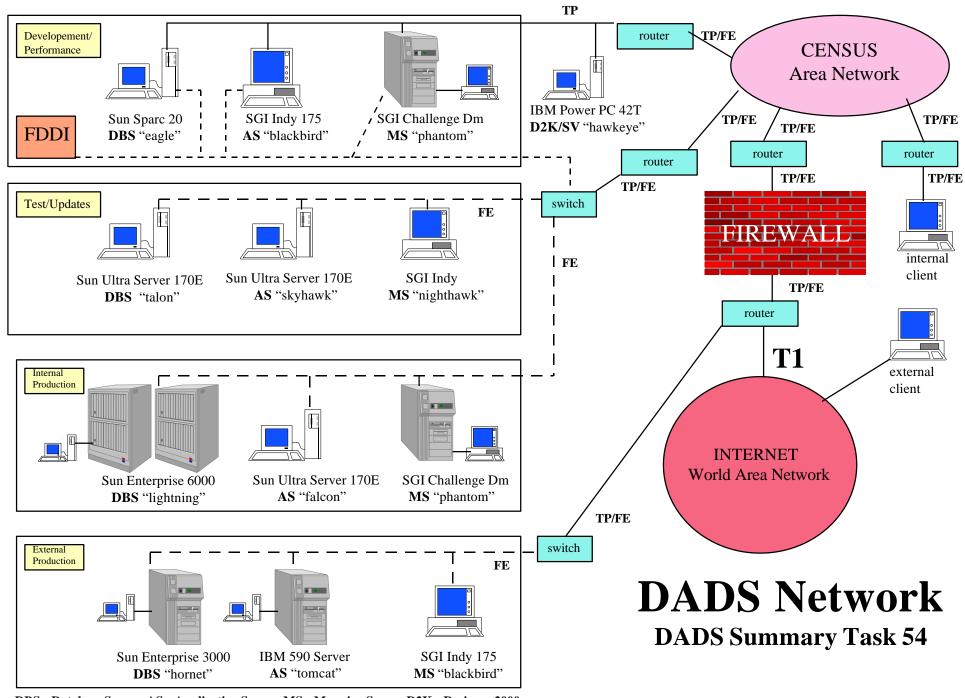
```
CHAR(1)
ur
                                                        NOT NULL,
air
                              CHAR(4)
                                                        NOT NULL,
geo_id
                              CHAR(11)
                                                        NOT NULL
STORAGE (
 INITIAL 134217728
 NEXT 134217728
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
 )
TABLESPACE &2
;
```

SPOOL OFF

```
NOT NULL,
 arelat1
                                   CHAR(1)
                                                               NOT NULL,
 asex
                                   CHAR(1)
                                   CHAR(1)
                                                               NOT NULL,
 arace
                                   CHAR(1)
                                                               NOT NULL,
 aage
 amarital
                                   CHAR(1)
                                                               NOT NULL,
                                                               NOT NULL
 ahispan
                                   CHAR(1)
STORAGE (
  INITIAL
            134217728
  NEXT 134217728
 MINEXTENTS 1
  MAXEXTENTS 99
  PCTINCREASE 0
TABLESPACE &2
;
REM
PROMPT
PROMPT Creating Table HOUSING100
CREATE TABLE housing100_&1 (
                                   CHAR(11)
                                                               NOT NULL,
house_id
 houstype
                                   CHAR(1)
                                                               NOT NULL,
 units1
                                   CHAR (2)
                                                               NOT NULL,
                                                               NOT NULL,
 rooms
                                   NUMBER (1)
                                                               NOT NULL,
 tenure
                                   CHAR(1)
 acreage
                                   CHAR(1)
                                                               NOT NULL,
                                                               NOT NULL,
 commuse
                                   CHAR(1)
 value
                                                               NOT NULL,
                                   CHAR(2)
                                                               NOT NULL,
 rent1
                                   CHAR(2)
 meals
                                   CHAR(1)
                                                               NOT NULL,
                                                               NOT NULL,
 vacancy1
                                   CHAR(1)
 vacancy2
                                   CHAR(1)
                                                               NOT NULL,
                                   CHAR(1)
                                                               NOT NULL,
 vacancy3
                                                               NOT NULL,
 vacancy4
                                   CHAR(1)
 persons
                                   NUMBER (5)
                                                               NOT NULL,
                                                               NOT NULL,
 perssrc
                                   CHAR(1)
 formtype
                                   CHAR(1)
                                                               NOT NULL,
                                                               NOT NULL,
 hsub
                                   CHAR(1)
 hpsub
                                   CHAR(1)
                                                               NOT NULL,
                                                               NOT NULL,
 hsda
                                   CHAR(1)
                                                               NOT NULL,
 gqtype
                                   CHAR(3)
 gqtypesc
                                   CHAR(1)
                                                               NOT NULL,
 gginst
                                   CHAR(1)
                                                               NOT NULL,
 gqinstsc
                                   CHAR(1)
                                                               NOT NULL,
                                                               NOT NULL,
 unit_id
                                   CHAR (7)
 state
                                   CHAR(2)
                                                               NOT NULL,
                                                               NOT NULL,
 county
                                   CHAR(3)
 tract
                                   CHAR(6)
                                                               NOT NULL,
                                                               NOT NULL,
 blkgroup
                                   CHAR(1)
                                                               NOT NULL,
                                   CHAR(2)
 block
                                                               NOT NULL,
                                   CHAR (5)
 zip
 msa
                                   CHAR (4)
                                                               NOT NULL,
 cmsa
                                   CHAR(2)
                                                               NOT NULL,
                                                               NOT NULL,
 place
                                   CHAR (4)
 mcd
                                   CHAR(3)
                                                               NOT NULL,
 cd101
                                   CHAR(2)
                                                               NOT NULL,
 cd103
                                   CHAR(2)
                                                               NOT NULL,
 anrc
                                   CHAR(2)
                                                               NOT NULL,
                                                               NOT NULL,
 นล
                                   CHAR (4)
 ur
                                   CHAR(1)
                                                               NOT NULL,
 air
                                   CHAR (4)
                                                               NOT NULL,
```

```
geo_id
                                 CHAR(11)
                                                             NOT NULL
)
STORAGE (
INITIAL 134217728
  NEXT 134217728
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
TABLESPACE &2
REM
PROMPT
PROMPT Creating Table PERSONS100
CREATE TABLE persons100_&1 (
pers_id
                                 CHAR(10)
                                                             NOT NULL,
house_id
                                  CHAR(11)
                                                             NOT NULL,
relat1
                                  CHAR(2)
                                                            NOT NULL,
                                                            NOT NULL,
                                 CHAR(1)
sex
race
                                                            NOT NULL,
                                 CHAR(3)
age
                                 NUMBER (3)
                                                             NOT NULL,
marital
                                 CHAR(1)
                                                             NOT NULL,
                                                             NOT NULL,
                                 CHAR(1)
hispanic
state
                                 CHAR(2)
                                                             NOT NULL,
county
                                 CHAR(3)
                                                             NOT NULL,
 tract
                                 CHAR(6)
                                                             NOT NULL,
                                                             NOT NULL,
blkgroup
                                 CHAR(1)
block
                                 CHAR(2)
                                                             NOT NULL,
                                 CHAR(5)
                                                             NOT NULL,
 zip
msa
                                  CHAR (4)
                                                             NOT NULL,
                                                             NOT NULL,
 cmsa
                                 CHAR(2)
                                                             NOT NULL,
place
                                 CHAR (4)
mcd
                                                             NOT NULL,
                                 CHAR(3)
cd101
                                 CHAR(2)
                                                             NOT NULL,
 cd103
                                 CHAR(2)
                                                             NOT NULL,
                                                             NOT NULL,
 anrc
                                 CHAR(2)
                                                             NOT NULL,
                                 CHAR (4)
ua
                                                             NOT NULL,
                                 CHAR(1)
ur
air
                                 CHAR(4)
                                                             NOT NULL,
                                                             NOT NULL
geo_id
                                 CHAR (11)
STORAGE (
 INITIAL 134217728
 NEXT 134217728
 MINEXTENTS 1
 MAXEXTENTS 99
 PCTINCREASE 0
TABLESPACE &2
SPOOL OFF
```

	#1	s for Prototype	rchitecture Diagr	Technical .	Appendix C:



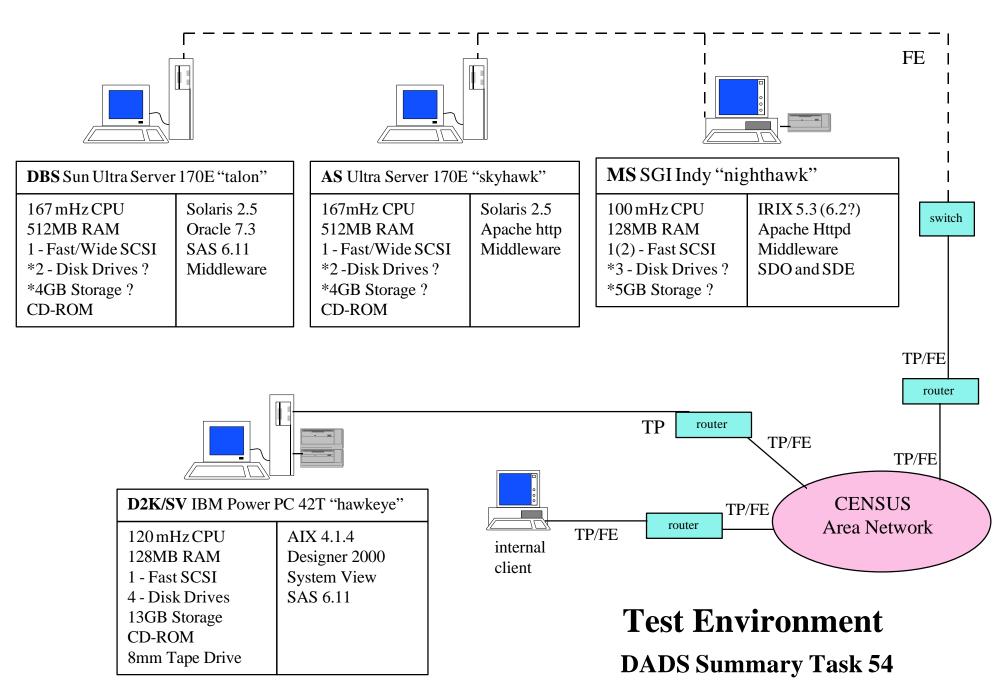
 $DBS = Database \ Server \ AS = Application \ Server \ MS = Mapping \ Server \ D2K = Designer \ 2000$ 

TP = Twisted Pair FE

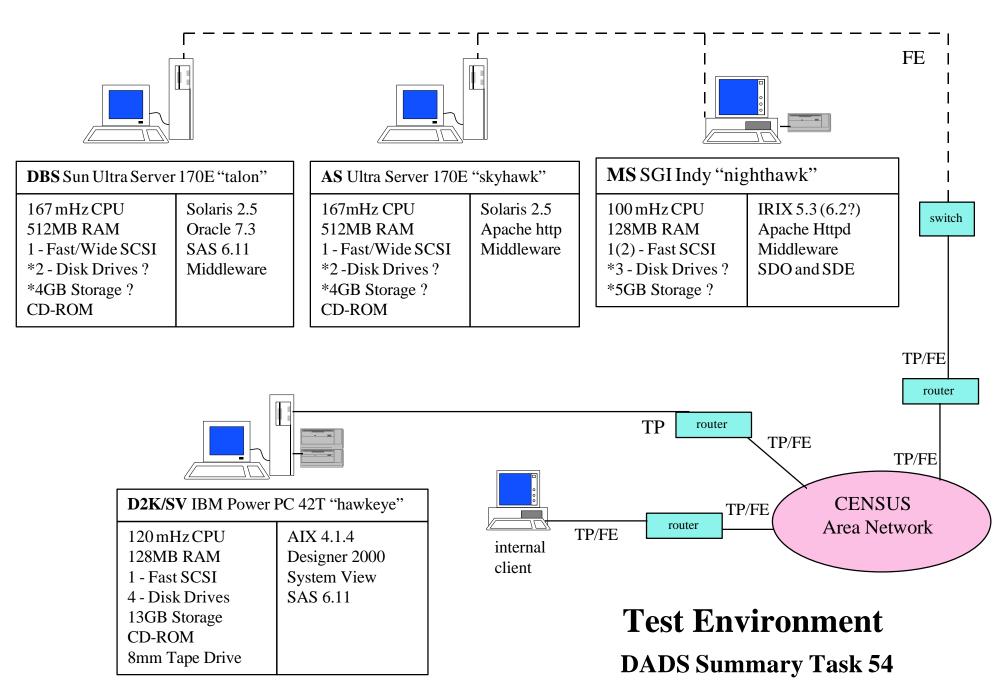
FE = Fast Ethernet

TI = Data Line

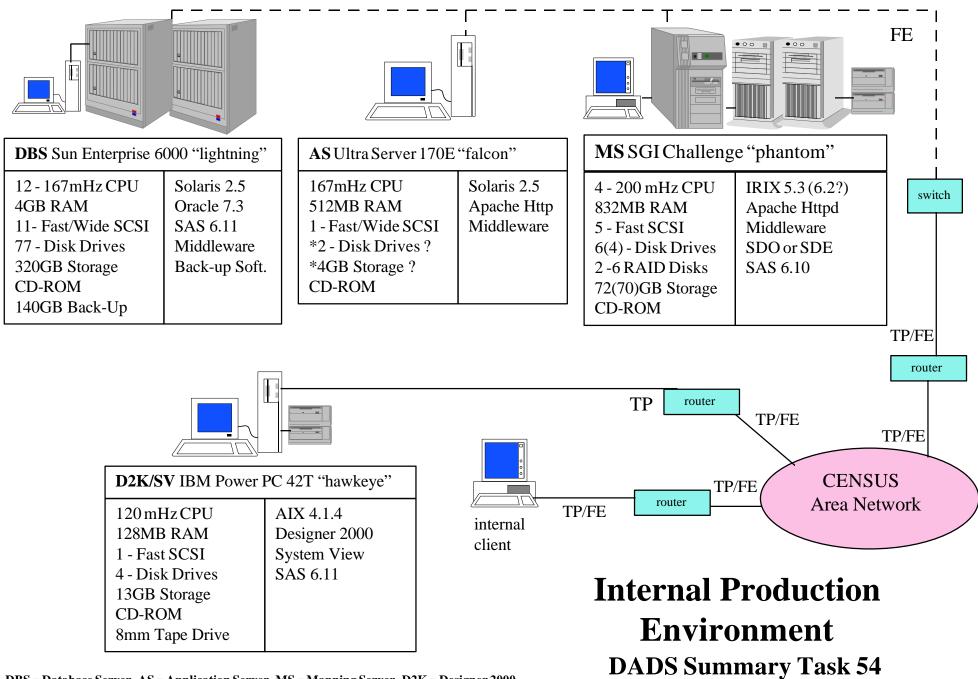
SV = System View



DBS = Database Server AS = Application Server MS = Mapping Server D2K = Designer 2000 SV = System View TP = Twisted Pair FE = Fast Ethernet



DBS = Database Server AS = Application Server MS = Mapping Server D2K = Designer 2000 SV = System View TP = Twisted Pair FE = Fast Ethernet



DBS = Database Server AS = Application Server MS = Mapping Server D2K = Designer 2000 SV = System View TP = Twisted Pair FE = Fast Ethernet

